OpenStack-Ansible Documentation: os_cinder role

Release 18.1.0.dev321

OpenStack-Ansible Contributors

Apr 17, 2024

CONTENTS

This Ansible role installs and configures OpenStack cinder.

The following cinder services are managed by the role:

- cinder-api
- cinder-volume
- cinder-scheduler

By default, cinder API v1 and v2 are both enabled.

CHAPTER

ONE

CONFIGURING THE BLOCK (CINDER) STORAGE SERVICE (OPTIONAL)

By default, the Block (cinder) storage service installs on the host itself using the LVM backend.

Note: While this is the default for cinder, using the LVM backend results in a Single Point of Failure.

The LVM back end needs to run on the host, however most of the other back ends can be deployed inside a container. If the storage back ends deployed within your environment are able to run inside containers, then it is recommended to set `is_metal: False` in the `env.d/cinder.yml` file.

Note: Due to a limitation of the container system, you must deploy the volume service directly onto the host when using back ends depending on iSCSI. That is the case, for example, for storage appliances configured to use the iSCSI protocol.

1.1 NFS backend

Edit `/etc/openstack_deploy/openstack_user_config.yml` and configure the NFS client on each storage node if the NetApp backend is configured to use an NFS storage protocol.

1. For each storage node, add one `cinder_backends` block underneath the a new `container_vars` section. `container_vars` are used to allow container/host individualized configuration. Each cinder back end is defined with a unique key. For example, `nfs-volume1`. This later represents a unique cinder backend and volume type.

   ```
   container_vars:
     cinder_backends:
       nfs-volume1:
   ```

2. Configure the appropriate cinder volume backend name:

   ```
   volume_backend_name: NFS_VOLUME1
   ```

3. Configure the appropriate cinder NFS driver:

   ```
   volume_driver: cinder.volume.drivers.nfs.NfsDriver
   ```

4. Configure the location of the file that lists shares available to the block storage service. This configuration file must include `nfs_shares_config`:

```
nfs_shares_config: FILENAME_NFS_SHARES
```

Replace FILENAME_NFS_SHARES with the location of the share configuration file. For example, /etc/cinder/nfs_shares_volume1.

5. Define mount options for the NFS mount. For example:

```
nfs_mount_options: "rsize=65535,wsize=65535,timeo=1200,actimeo=120"
```

6. Configure one or more NFS shares:

```
shares:
   - { ip: "HOSTNAME", share: "PATH_TO_NFS_VOLUME" }
```

Replace HOSTNAME with the IP address or hostname of the NFS server, and the PATH_TO_NFS_VOLUME with the absolute path to an existing and accessible NFS share (excluding the IP address or hostname).

The following is a full configuration example of a cinder NFS backend named NFS1. The cinder playbooks will automatically add a custom volume-type and nfs-volume1 as in this example:

```
container_vars:
  cinder_backends:
    nfs-volume1:
      volume_backend_name: NFS_VOLUME1
      volume_driver: cinder.volume.drivers.nfs.NfsDriver
      nfs_shares_config: /etc/cinder/nfs_shares_volume1
      nfs_mount_options: "rsize=65535,wsize=65535,timeo=1200,
↪actimeo=120"
      shares:
      - { ip: "1.2.3.4", share: "/vol1" }
```

1.2 Backup

You can configure cinder to backup volumes to Object Storage (swift). Enable the default configuration to back up volumes to a swift installation accessible within your environment. Alternatively, you can set cinder_service_backup_swift_url and other variables to back up to an external swift installation.

1. Add or edit the following line in the /etc/openstack_deploy/user_variables.yml file and set the value to True:

```
cinder_service_backup_program_enabled: True
```

2. By default, cinder uses the access credentials of the user initiating the backup. Default values are set in the /opt/openstack-ansible/playbooks/roles/os_cinder/defaults/main.yml file. You can override those defaults by setting variables in /etc/openstack_deploy/user_variables.yml to change how cinder performs backups. Add and edit any of the following variables to the /etc/openstack_deploy/user_variables.yml file:

```
...
cinder_service_backup_swift_auth: per_user
```

(continues on next page)

(continued from previous page)

```
# Options include 'per_user' or 'single_user'. We default to
# 'per_user' so that backups are saved to a user's swift
# account.
cinder_service_backup_swift_url:
# This is your swift storage url when using 'per_user', or keystone
# endpoint when using 'single_user'.  When using 'per_user', you
# can leave this as empty or as None to allow cinder-backup to
# obtain a storage url from environment.
cinder_service_backup_swift_url:
cinder_service_backup_swift_auth_version: 2
cinder_service_backup_swift_user:
cinder_service_backup_swift_tenant:
cinder_service_backup_swift_key:
cinder_service_backup_swift_container: volumebackups
cinder_service_backup_swift_object_size: 52428800
cinder_service_backup_swift_retry_attempts: 3
cinder_service_backup_swift_retry_backoff: 2
cinder_service_backup_compression_algorithm: zlib
cinder_service_backup_metadata_version: 2
```

During installation of cinder, the backup service is configured.

1.3 Using Ceph for cinder backups

You can deploy Ceph to hold cinder volume backups. To get started, set the `cinder_service_backup_driver` Ansible variable:

```
cinder_service_backup_driver: cinder.backup.drivers.ceph.CephBackupDriver
```

Configure the Ceph user and the pool to use for backups. The defaults are shown here:

```
cinder_service_backup_ceph_user: cinder-backup
cinder_service_backup_ceph_pool: backups
```

1.4 Availability zones

Create multiple availability zones to manage cinder storage hosts. Edit the `/etc/openstack_deploy/openstack_user_config.yml` and `/etc/openstack_deploy/user_variables.yml` files to set up availability zones.

1. For each cinder storage host, configure the availability zone under the `container_vars` stanza:

   ```
   cinder_storage_availability_zone: CINDERAZ
   ```

   Replace `CINDERAZ` with a suitable name. For example `cinderAZ_2`.

2. If more than one availability zone is created, configure the default availability zone for all the hosts by creating a `cinder_default_availability_zone` in your `/etc/openstack_deploy/user_variables.yml`

```
cinder_default_availability_zone: CINDERAZ_DEFAULT
```

Replace CINDERAZ_DEFAULT with a suitable name. For example, cinderAZ_1. The default availability zone should be the same for all cinder hosts.

1.5 OpenStack Dashboard (horizon) configuration for cinder

You can configure variables to set the behavior for cinder volume management in OpenStack Dashboard (horizon). By default, no horizon configuration is set.

1. The default destination availability zone is nova if you use multiple availability zones and cinder_default_availability_zone has no definition. Volume creation with horizon might fail if there is no availability zone named nova. Set cinder_default_availability_zone to an appropriate availability zone name so that *Any availability zone* works in horizon.
2. horizon does not populate the volume type by default. On the new volume page, a request for the creation of a volume with the default parameters fails. Set cinder_default_volume_type so that a volume creation request without an explicit volume type succeeds.

1.6 Configuring cinder to use LVM

1. List the container_vars that contain the storage options for the target host.

Note: The vars related to the cinder availability zone and the limit_container_types are optional.

To configure an LVM, utilize the following example:

```
storage_hosts:
 Infra01:
   ip: 172.29.236.16
   container_vars:
     cinder_storage_availability_zone: cinderAZ_1
     cinder_default_availability_zone: cinderAZ_1
     cinder_backends:
       lvm:
         volume_backend_name: LVM_iSCSI
         volume_driver: cinder.volume.drivers.lvm.LVMVolumeDriver
         volume_group: cinder-volumes
         iscsi_ip_address: "{{ cinder_storage_address }}"
       limit_container_types: cinder_volume
```

To use another backend in a container instead of bare metal, copy the env.d/cinder.yml to /etc/openstack_deploy/env.d/cinder.yml file and change the is_metal: true stanza under the cinder_volumes_container properties to is_metal: false.

Alternatively, you can also selectively override, like this:

```
container_skel:
  cinder_volumes_container:
    properties:
      is_metal: false
```

1.7 Configuring cinder to use Ceph

In order for cinder to use Ceph, it is necessary to configure for both the API and backend. When using any forms of network storage (iSCSI, NFS, Ceph) for cinder, the API containers can be considered as backend servers. A separate storage host is not required.

Copy the `env.d/cinder.yml` to `/etc/openstack_deploy/env.d/cinder.yml` file and change the `is_metal: true` stanza under the `cinder_volumes_container` properties to `is_metal: false`.

Alternatively, you can also selectively override, like this:

```
container_skel:
  cinder_volumes_container:
    properties:
      is_metal: false
```

1. List of target hosts on which to deploy the cinder API. We recommend that a minimum of three target hosts are used for this service.

   ```
   storage-infra_hosts:
     infra1:
       ip: 172.29.236.101
     infra2:
       ip: 172.29.236.102
     infra3:
       ip: 172.29.236.103
   ```

   To configure an RBD backend, utilize the following example:

   ```
   container_vars:
     cinder_storage_availability_zone: cinderAZ_3
     cinder_default_availability_zone: cinderAZ_1
     cinder_backends:
       limit_container_types: cinder_volume
       rbd_backend:
         volume_driver: cinder.volume.drivers.rbd.RBDDriver
         rbd_pool: volumes
         rbd_ceph_conf: /etc/ceph/ceph.conf
         rbd_flatten_volume_from_snapshot: 'false'
         rbd_max_clone_depth: 5
         rbd_store_chunk_size: 4
         rados_connect_timeout: 30
         volume_backend_name: rbd_backend
         rbd_user: "{{ cinder_ceph_client }}"
         rbd_secret_uuid: "{{ cinder_ceph_client_uuid }}"
   ```

The following example sets cinder to use the `cinder_volumes` pool. The example uses cephx authentication and requires existing `cinder` account for `cinder_volumes` pool.

In `user_variables.yml`:

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

In `openstack_user_config.yml`:

```
storage_hosts:
 infra1:
  ip: 172.29.236.101
  container_vars:
   cinder_backends:
     limit_container_types: cinder_volume
     rbd:
       volume_group: cinder-volumes
       volume_driver: cinder.volume.drivers.rbd.RBDDriver
       volume_backend_name: rbd
       rbd_pool: cinder-volumes
       rbd_ceph_conf: /etc/ceph/ceph.conf
       rbd_user: cinder
 infra2:
  ip: 172.29.236.102
  container_vars:
   cinder_backends:
     limit_container_types: cinder_volume
     rbd:
       volume_group: cinder-volumes
       volume_driver: cinder.volume.drivers.rbd.RBDDriver
       volume_backend_name: rbd
       rbd_pool: cinder-volumes
       rbd_ceph_conf: /etc/ceph/ceph.conf
       rbd_user: cinder
 infra3:
  ip: 172.29.236.103
  container_vars:
   cinder_backends:
     limit_container_types: cinder_volume
     rbd:
       volume_group: cinder-volumes
       volume_driver: cinder.volume.drivers.rbd.RBDDriver
       volume_backend_name: rbd
       rbd_pool: cinder-volumes
       rbd_ceph_conf: /etc/ceph/ceph.conf
       rbd_user: cinder
```

This link provides a complete working example of Ceph setup and integration with cinder (nova and glance included):

- OpenStack-Ansible and Ceph Working Example

1.8 Configuring cinder to use Dell EqualLogic

To use the Dell EqualLogic volume driver as a back end, edit the `/etc/openstack_deploy/openstack_user_config.yml` file and configure the storage nodes that will use it.

Define the following parameters.

1. Add `dellqlx` stanza under the `cinder_backends` for each storage node:

```
cinder_backends:
  delleqlx:
```

2. Specify volume back end name:

```
volume_backend_name: DellEQLX_iSCSI
```

3. Use Dell EQLX San ISCSI driver:

```
volume_driver: cinder.volume.drivers.dell_emc.ps.PSSeriesISCSIDriver
```

4. Specify the SAN IP address:

```
san_ip: ip_of_dell_storage
```

5. Specify SAN username (Default: grpadmin):

```
san_login: grpadmin
```

6. Specify the SAN password:

```
san_password: password
```

7. Specify the group name for pools (Default: group-0):

```
eqlx_group_name: group-0
```

8. Specify the pool where Cinder will create volumes and snapshots (Default: default):

```
eqlx_pool: default
```

9. Ensure the `openstack_user_config.yml` configuration is accurate:

```
storage_hosts:
  Infra01:
    ip: infra_host_ip
    container_vars:
      cinder_backends:
        limit_container_types: cinder_volume
        delleqlx:
          volume_backend_name: DellEQLX_iSCSI
          volume_driver: cinder.volume.drivers.dell_emc.ps.
↪PSSeriesISCSIDriver
          san_ip: ip_of_dell_storage
          san_login: grpadmin
          san_password: password
          eqlx_group_name: group-0
          eqlx_pool: default
```

Note: For more details about available configuration options, see https://docs.openstack.org/ocata/config-reference/block-storage/drivers/dell-equallogic-driver.html

1.9 Configuring cinder to use a NetApp appliance

To use a NetApp storage appliance back end, edit the `/etc/openstack_deploy/openstack_user_config.yml` file and configure each storage node that will use it.

Note: Ensure that the NAS Team enables `httpd.admin.access`.

1. Add the `netapp` stanza under the `cinder_backends` stanza for each storage node:

   ```
   cinder_backends:
     netapp:
   ```

   The options in subsequent steps fit under the `netapp` stanza.

   The backend name is arbitrary and becomes a volume type within cinder.

2. Configure the storage family:

   ```
   netapp_storage_family: STORAGE_FAMILY
   ```

   Replace `STORAGE_FAMILY` with `ontap_7mode` for Data ONTAP operating in 7-mode or `ontap_cluster` for Data ONTAP operating as a cluster.

3. Configure the storage protocol:

   ```
   netapp_storage_protocol: STORAGE_PROTOCOL
   ```

   Replace `STORAGE_PROTOCOL` with `iscsi` for iSCSI or `nfs` for NFS.

   For the NFS protocol, specify the location of the configuration file that lists the shares available to cinder:

   ```
   nfs_shares_config: FILENAME_NFS_SHARES
   ```

   Replace `FILENAME_NFS_SHARES` with the location of the share configuration file. For example, `/etc/cinder/nfs_shares`.

4. Configure the server:

   ```
   netapp_server_hostname: SERVER_HOSTNAME
   ```

   Replace `SERVER_HOSTNAME` with the hostnames for both netapp controllers.

5. Configure the server API port:

   ```
   netapp_server_port: PORT_NUMBER
   ```

   Replace `PORT_NUMBER` with 80 for HTTP or 443 for HTTPS.

6. Configure the server credentials:

```
netapp_login: USER_NAME
netapp_password: PASSWORD
```

Replace USER_NAME and PASSWORD with the appropriate values.

7. Select the NetApp driver:

```
volume_driver: cinder.volume.drivers.netapp.common.NetAppDriver
```

8. Configure the volume back end name:

```
volume_backend_name: BACKEND_NAME
```

Replace BACKEND_NAME with a value that provides a hint for the cinder scheduler. For example, NETAPP_iSCSI.

9. Ensure the openstack_user_config.yml configuration is accurate:

```
storage_hosts:
  Infra01:
    ip: 172.29.236.16
    container_vars:
      cinder_backends:
        limit_container_types: cinder_volume
        netapp:
          netapp_storage_family: ontap_7mode
          netapp_storage_protocol: nfs
          netapp_server_hostname: 111.222.333.444
          netapp_server_port: 80
          netapp_login: openstack_cinder
          netapp_password: password
          volume_driver: cinder.volume.drivers.netapp.common.
↪NetAppDriver
          volume_backend_name: NETAPP_NFS
```

For netapp_server_hostname, specify the IP address of the Data ONTAP server. Include iSCSI or NFS for the netapp_storage_family depending on the configuration. Add 80 if using HTTP or 443 if using HTTPS for netapp_server_port.

The cinder-volume.yml playbook will automatically install the nfs-common file across the hosts, transitioning from an LVM to a NetApp back end.

1.10 Configuring cinder qos specs

Deployers may optionally define the variable cinder_qos_specs to create qos specs. This variable is a list of dictionaries that contain the options for each qos spec. cinder volume-types may be assigned to a qos spec by defining the key cinder_volume_types in the desired qos spec dictionary.

```
- name: high-iops
  options:
    consumer: front-end
    read_iops_sec: 2000
    write_iops_sec: 2000
  cinder_volume_types:
```

(continues on next page)

(continued from previous page)

```
    - volumes-1
    - volumes-2
- name: low-iops
  options:
    consumer: front-end
    write_iops_sec: 100
```

1.11 Configure cinder to create a private volume type

By default, a define cinder-backend will create a public volume type. In order to create a private volume type, add the variable `public` and set it to `false`. If it is set to `true` or not set at all the volume type created will be public.

```
cinder_backends:
  private_type:
    public: false
```

1.12 Shared storage and synchronized UID/GID

Specify a custom UID for the cinder user and GID for the cinder group to ensure they are identical on each host. This is helpful when using shared storage on Compute nodes because it allows instances to migrate without filesystem ownership failures.

By default, Ansible creates the cinder user and group without specifying the UID or GID. To specify custom values for the UID or GID, set the following Ansible variables:

```
cinder_system_user_uid = <specify a UID>
cinder_system_group_gid = <specify a GID>
```

Warning: Setting this value after deploying an environment with OpenStack-Ansible can cause failures, errors, and general instability. These values should only be set once before deploying an OpenStack environment and then never changed.

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

CHAPTER

TWO

DEFAULT VARIABLES

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

# Set the host which will execute the shade modules
# for the service setup. The host must already have
# clouds.yaml properly configured.
cinder_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
cinder_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((cinder_service_setup_host ==
↪'localhost') | ternary(ansible_playbook_python, ansible_facts['python'][
↪'executable'])) }}"

# Set installation method.
cinder_install_method: "{{ service_install_method | default('source') }}"
cinder_venv_python_executable: "{{ openstack_venv_python_executable |␣
↪default('python3') }}"

cinder_git_repo: https://opendev.org/openstack/cinder
cinder_git_install_branch: master
cinder_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
cinder_git_constraints:
  - "--constraint {{ cinder_upper_constraints_url }}"

cinder_pip_install_args: "{{ pip_install_options | default('') }}"

# Name of the virtual env to deploy into
cinder_venv_tag: "{{ venv_tag | default('untagged') }}"
cinder_bin: "{{ _cinder_bin }}"

# Enable/Disable Barbican
cinder_barbican_enabled: "{{ (groups['barbican_all'] is defined) and␣
↪(groups['barbican_all'] | length > 0) }}"

# Enable/Disable Ceilometer
cinder_ceilometer_enabled: "{{ (groups['ceilometer_all'] is defined) and␣
↪(groups['ceilometer_all'] | length > 0) }}"

# Time period for which to generate volume usages. The options are hour,␣
↪day,
```

(continues on next page)

(continued from previous page)

```
# month, or year. (string value)
cinder_volume_usage_audit: hour
cinder_volume_usage_audit_send_actions_enabled: True

cinder_storage_availability_zone: nova
cinder_default_availability_zone: "{{ cinder_storage_availability_zone }}"

cinder_storage_address: 127.0.0.1
cinder_management_address: 127.0.0.1
cinder_uwsgi_bind_address: "{{ openstack_service_bind_address | default('0.
↪0.0.0') }}"

cinder_nova_catalog_info: compute:nova:internalURL
cinder_nova_catalog_admin_info: compute:nova:adminURL

cinder_fatal_deprecations: False

## Database info
cinder_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}
↪"
cinder_db_setup_python_interpreter: "{{ openstack_db_setup_python_
↪interpreter | default((cinder_db_setup_host == 'localhost') |␣
↪ternary(ansible_playbook_python, ansible_facts['python']['executable']))␣
↪}}"
cinder_galera_address: "{{ galera_address | default('127.0.0.1') }}"
cinder_galera_user: cinder
cinder_galera_database: cinder
cinder_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
cinder_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('/etc/ssl/
↪certs/galera-ca.pem') }}"
cinder_galera_port: "{{ galera_port | default('3306') }}"

## Oslo Messaging

# RabbitMQ
cinder_oslomsg_heartbeat_in_pthread: "{{ oslomsg_heartbeat_in_pthread |␣
↪default(_cinder_oslomsg_heartbeat_in_pthread) }}"

# RPC
cinder_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default(
↪'rabbitmq_all') }}"
cinder_oslomsg_rpc_setup_host: "{{ (cinder_oslomsg_rpc_host_group in␣
↪groups) | ternary(groups[cinder_oslomsg_rpc_host_group][0], 'localhost')␣
↪}}"
cinder_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit
↪') }}"
cinder_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1')␣
↪}}"
cinder_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
cinder_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
cinder_oslomsg_rpc_userid: cinder
cinder_oslomsg_rpc_vhost: /cinder
cinder_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
↪'TLSv1_2') }}"
cinder_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }
↪}"
```

(continues on next page)

(continued from previous page)

```
# Notify
cinder_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
cinder_oslomsg_notify_setup_host: "{{ (cinder_oslomsg_notify_host_group in␣
↪groups) | ternary(groups[cinder_oslomsg_notify_host_group][0], 'localhost
↪') }}"
cinder_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
cinder_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.
↪0.1') }}"
cinder_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
cinder_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False)␣
↪}}"
cinder_oslomsg_notify_userid: "{{ cinder_oslomsg_rpc_userid }}"
cinder_oslomsg_notify_password: "{{ cinder_oslomsg_rpc_password }}"
cinder_oslomsg_notify_vhost: "{{ cinder_oslomsg_rpc_vhost }}"
cinder_oslomsg_notify_ssl_version: "{{ oslomsg_notify_ssl_version |␣
↪default('TLSv1_2') }}"
cinder_oslomsg_notify_ssl_ca_file: "{{ oslomsg_notify_ssl_ca_file |␣
↪default('') }}"

## (Qdrouterd) integration
# TODO(evrardjp): Change structure when more backends will be supported
cinder_oslomsg_amqp1_enabled: "{{ cinder_oslomsg_rpc_transport == 'amqp' }}
↪"

## Cinder User / Group
cinder_system_user_name: cinder
cinder_system_group_name: cinder
cinder_system_comment: cinder system user
cinder_system_shell: /bin/false
cinder_system_home_folder: "/var/lib/{{ cinder_system_user_name }}"

## Manually specified cinder UID/GID
# Deployers can specify a UID for the cinder user as well as the GID for␣
↪the
# cinder group if needed. This is commonly used in environments where␣
↪shared
# storage is used, such as NFS or GlusterFS, and cinder UID/GID values␣
↪must be
# in sync between multiple servers.
#
# WARNING: Changing these values on an existing deployment can lead to
#          failures, errors, and instability.
#
# cinder_system_user_uid = <UID>
# cinder_system_group_gid = <GID>

cinder_lock_path: /var/lock/cinder

## Cinder Auth
cinder_service_admin_tenant_name: "service"
cinder_service_admin_username: "cinder"

## Cinder API's enabled
```

(continues on next page)

(continued from previous page)

```
cinder_enable_v2_api: false

## Cinder api service type and data
cinder_service_name: cinder
cinder_service_project_domain_id: default
cinder_service_user_domain_id: default
cinder_service_user_name: cinder
cinder_service_project_name: service
cinder_service_role_name: admin
cinder_service_region: "{{ service_region | default('RegionOne') }}"
cinder_service_description: "Cinder Volume Service"
cinder_service_port: 8776
cinder_service_proto: http
cinder_service_publicuri_proto: "{{ openstack_service_publicuri_proto |␣
↪default(cinder_service_proto) }}"
cinder_service_adminuri_proto: "{{ openstack_service_adminuri_proto |␣
↪default(cinder_service_proto) }}"
cinder_service_internaluri_proto: "{{ openstack_service_internaluri_proto␣
↪| default(cinder_service_proto) }}"
cinder_service_type: volume
cinder_service_publicuri: "{{ cinder_service_publicuri_proto }}://{{␣
↪external_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_adminuri: "{{ cinder_service_adminuri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_internaluri: "{{ cinder_service_internaluri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ cinder_service_port }}"

cinder_service_v2_name: cinderv2
cinder_service_v2_port: 8776
cinder_service_v2_proto: http
cinder_service_v2_type: volumev2
cinder_service_v2_description: "Cinder Volume Service V2"
cinder_service_v2_publicuri: "{{ cinder_service_publicuri_proto }}://{{␣
↪external_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_v2_publicurl: "{{ cinder_service_publicuri }}/v2/%(tenant_
↪id)s"
cinder_service_v2_adminuri: "{{ cinder_service_adminuri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_v2_adminurl: "{{ cinder_service_adminuri }}/v2/%(tenant_id)s
↪"
cinder_service_v2_internaluri: "{{ cinder_service_internaluri_proto }}://{
↪{ internal_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_v2_internalurl: "{{ cinder_service_internaluri }}/v2/
↪%(tenant_id)s"

cinder_service_v3_name: cinderv3
cinder_service_v3_port: 8776
cinder_service_v3_proto: http
cinder_service_v3_type: volumev3
cinder_service_v3_description: "Cinder Volume Service V3"
cinder_service_v3_publicuri: "{{ cinder_service_publicuri_proto }}://{{␣
↪external_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_v3_publicurl: "{{ cinder_service_publicuri }}/v3/%(tenant_
↪id)s"
cinder_service_v3_adminuri: "{{ cinder_service_adminuri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ cinder_service_port }}"
```

(continues on next page)

(continued from previous page)

```
cinder_service_v3_adminurl: "{{ cinder_service_adminuri }}/v3/%(tenant_id)s
↪"
cinder_service_v3_internaluri: "{{ cinder_service_internaluri_proto }}://{
↪{ internal_lb_vip_address }}:{{ cinder_service_port }}"
cinder_service_v3_internalurl: "{{ cinder_service_internaluri }}/v3/
↪%(tenant_id)s"

cinder_auth_strategy: keystone

## Keystone authentication middleware
cinder_keystone_auth_plugin: "{{ cinder_keystone_auth_type }}"
cinder_keystone_auth_type: password

## In order to enable the cinder backup you MUST set ``cinder_service_
↪backup_program_enabled`` to "true"
cinder_service_backup_program_enabled: false
# cinder_service_backup_driver: Options include 'cinder.backup.drivers.
↪swift.SwiftBackupDriver' or
#                               'cinder.backup.drivers.ceph.
↪CephBackupDriver'
cinder_service_backup_driver: cinder.backup.drivers.swift.SwiftBackupDriver
# cinder_service_backup_swift_auth: Options include 'per_user' or 'single_
↪user', we default to
#                                   'per_user' so that backups are saved␣
↪to a user's swift account.
cinder_service_backup_swift_auth: per_user
# cinder_service_backup_swift_url: This is your swift storage url when␣
↪using 'per_user', or keystone
#                                  endpoint when using 'single_user'. ␣
↪When using 'per_user', you
#                                  can leave this as empty or as None to␣
↪allow cinder-backup to
#                                  obtain storage url from environment.
cinder_service_backup_swift_url:
cinder_service_backup_swift_auth_version: 2
cinder_service_backup_swift_user:
cinder_service_backup_swift_tenant:
cinder_service_backup_swift_key:
cinder_service_backup_swift_container: volumebackups
cinder_service_backup_swift_object_size: 52428800
cinder_service_backup_swift_retry_attempts: 3
cinder_service_backup_swift_retry_backoff: 2
cinder_service_backup_ceph_user: cinder-backup
cinder_service_backup_ceph_pool: backups
cinder_service_backup_compression_algorithm: zlib
cinder_service_backup_metadata_version: 2

cinder_swift_catalog_info: "object-store:swift:internalURL"

## Cap the maximun number of threads / workers when a user value is␣
↪unspecified.
cinder_osapi_volume_workers_max: 16
cinder_osapi_volume_workers: "{{ [[(ansible_facts['processor_vcpus']//
↪ansible_facts['processor_threads_per_core'])|default(1), 1] | max * 2,␣
↪cinder_osapi_volume_workers_max] | min }}"
```

(continues on next page)

(continued from previous page)

```
## Cinder iscsi
cinder_target_helper_mapping:
  RedHat: lioadm
  Debian: tgtadm
cinder_target_helper: "{{ cinder_target_helper_mapping[ansible_facts['os_
↪family']] }}"
cinder_iscsi_iotype: fileio
cinder_iscsi_num_targets: 100
cinder_iscsi_port: 3260

## Cinder RPC
cinder_rpc_executor_thread_pool_size: 64
cinder_rpc_response_timeout: 60

# (StrOpt) Method used to wipe old volumes (valid options are: none, zero)
cinder_volume_clear: zero
# (StrOpt) The flag to pass to ionice to alter the i/o priority of the␣
↪process
# used to zero a volume after deletion, for example "-c3" for idle only
# priority.
cinder_volume_clear_ionice: -c3

# (IntOpt) Size in MiB to wipe at start of old volumes. 0 => all
cinder_volume_clear_size: 0

cinder_volume_name_template: volume-%s

# osprofiler
cinder_profiler_enabled: false
# cinder_profiler_hmac_key is set in user_secrets.yml
cinder_profiler_trace_sqlalchemy: false

cinder_client_socket_timeout: 900

## Cinder quota
cinder_quota_volumes: 10
cinder_quota_snapshots: 10
cinder_quota_consistencygroups: 10
cinder_quota_gigabytes: 1000
cinder_quota_backups: 10
cinder_quota_backup_gigabytes: 1000

## General configuration
# cinder_backends:
#   lvm:
#     volume_group: cinder-volumes
#     volume_driver: cinder.volume.drivers.lvm.LVMVolumeDriver
#     volume_backend_name: LVM_iSCSI
#     extra_volume_types:
#       - lvm_high_iops
#       - lvm_low_iops

# Override generated device filter in lvm.conf
# Example:
# cinder_lvm_devices_filter_override:
#  - '"a/sd.*/"'
```

(continues on next page)

(continued from previous page)

```
#   - '"a/hd.*/"'
cinder_lvm_devices_filter_override: []

# cinder_backend_lvm_inuse: True if current host has an lvm backend
cinder_backend_lvm_inuse: '{{ (cinder_backends|default("")|to_json).find(
↪"lvm") != -1 }}'
# cinder_backend_rbd_inuse: True if the current host has an rbd backend
cinder_backend_rbd_inuse: '{{ (cinder_backends|default("")|to_json).find(
↪"cinder.volume.drivers.rbd.RBDDriver") != -1 }}'

# Set to false if you want to explicitly disable active/active cluster
cinder_active_active_cluster: "{{ cinder_backend_rbd_inuse }}"
cinder_active_active_cluster_name: ceph

## Policy vars
# Provide a list of access controls to merge with the default
# access controls in the service code.
#cinder_policy_overrides:
#   "volume:create": ""
#   "volume:delete": ""

# Comma separated list of Glance API servers
cinder_glance_api_servers: "{{ (glance_service_internalurl | default(
↪'http://localhost')) | urlsplit('scheme') ~ '://' ~ (glance_service_
↪internalurl | default('http://localhost')) | urlsplit('netloc') }}"

cinder_service_in_ldap: "{{ service_ldap_backend_enabled | default(False) }
↪}"

# Common pip packages
cinder_pip_packages:
  - "git+{{ cinder_git_repo }}@{{ cinder_git_install_branch }}#egg=cinder"
  - cryptography
  - ecdsa
  - httplib2
  - keystonemiddleware
  - osprofiler
  - PyMySQL
  - pymemcache
  - python-memcached
  - systemd-python

# Specific pip packages provided by the user
cinder_user_pip_packages: []

cinder_optional_oslomsg_amqp1_pip_packages:
  - oslo.messaging[amqp1]

cinder_api_init_overrides: {}
cinder_scheduler_init_overrides: {}
cinder_volume_init_overrides: {}
cinder_backup_init_overrides: {}

## Service Name-Group Mapping
cinder_services:
  cinder-scheduler:
```

(continues on next page)

(continued from previous page)

```
    group: cinder_scheduler
    service_name: cinder-scheduler
    init_config_overrides: "{{ cinder_scheduler_init_overrides }}"
    start_order: 1
    execstarts: "{{ cinder_bin }}/cinder-scheduler"
    execreloads: "/bin/kill -HUP $MAINPID"
  cinder-volume:
    group: cinder_volume
    service_name: cinder-volume
    init_config_overrides: "{{ cinder_volume_init_overrides }}"
    start_order: 2
    execstarts: "{{ cinder_bin }}/cinder-volume"
    execreloads: "/bin/kill -HUP $MAINPID"
  cinder-backup:
    group: cinder_backup
    service_name: cinder-backup
    condition: "{{ cinder_service_backup_program_enabled | bool }}"
    init_config_overrides: "{{ cinder_backup_init_overrides }}"
    start_order: 3
    execstarts: "{{ cinder_bin }}/cinder-backup"
    execreloads: "/bin/kill -HUP $MAINPID"
  cinder-api:
    group: cinder_api
    service_name: cinder-api
    init_config_overrides: "{{ cinder_api_init_overrides }}"
    start_order: 4
    wsgi_app: True
    wsgi_name: cinder-wsgi
    uwsgi_overrides: "{{ cinder_api_uwsgi_ini_overrides }}"
    uwsgi_port: "{{ cinder_service_port }}"
    uwsgi_bind_address: "{{ cinder_uwsgi_bind_address }}"

# Cinder uWSGI settings
cinder_wsgi_processes_max: 16
cinder_wsgi_processes: "{{ [[ansible_facts['processor_vcpus']|default(1),␣
↪1] | max * 2, cinder_wsgi_processes_max] | min }}"
cinder_wsgi_threads: 1

# Define the following dictionary variable to enable qos settings on␣
↪volumes.
# cinder_qos_specs
# - name: high-iops
#   options:
#     consumer: front-end
#     read_iops_sec: 2000
#     write_iops_sec: 2000
#   cinder_volume_types:
#     - volumes-1
#     - volumes-2
# - name: low-iops
#   options:
#     consumer: front-end
#     write_iops_sec: 100

## Tunable overrides
cinder_policy_overrides: {}
```

(continues on next page)

(continued from previous page)

```
cinder_rootwrap_conf_overrides: {}
cinder_api_paste_ini_overrides: {}
cinder_cinder_conf_overrides: {}
cinder_api_uwsgi_ini_overrides: {}
cinder_resource_filters_overrides: {}

## Set default cinder path in service units. The default override sets the
## execution path for the cinder service.
cinder_environment_overrides:
  Service:
    Environment: "PATH={{ cinder_bin }}:/usr/local/sbin:/usr/local/bin:/
↪usr/sbin:/usr/bin:/sbin:/bin"

_UUID_regex: "[0-9a-f]{8}-([0-9a-f]{4}-){3}[0-9a-f]{12}"

cinder_memcached_servers: "{{ memcached_servers }}"
```

CHAPTER

THREE

DEPENDENCIES

This role needs pip >= 7.1 installed on the target host.

This role needs to have the following variables defined:

```
# Comma separated list of Glance API servers
cinder_glance_api_servers: "http://glance_server:9292"

# Hostname or IP address of the Galera database
cinder_galera_address: "1.2.3.4"
```

This list is not exhaustive at present. See role internals for further details.

CHAPTER

FOUR

EXAMPLE PLAYBOOK

```yaml
---
- name: Installation and setup of cinder
  hosts: cinder_all
  user: root
  roles:
    - { role: "os_cinder", tags: [ "os-cinder" ] }
  vars:
    cinder_glance_api_servers: "http://glance_server:9292"
    cinder_galera_address: "{{ internal_lb_vip_address }}"
    galera_root_user: root
  vars_prompt:
    - name: "galera_root_password"
      prompt: "What is galera_root_password?"
```

CHAPTER

FIVE

EXTERNAL RESTART HOOKS

When the role performs a restart of the service, it will notify an Ansible handler named `Manage LB`, which is a noop within this role. In the playbook, other roles may be loaded before and after this role which will implement Ansible handler listeners for `Manage LB`, allowing external roles to manage the load balancer endpoints responsible for sending traffic to the servers being restarted by marking them in maintenance or active mode, draining sessions, etc. For an example implementation, please reference the ansible-haproxy-endpoints role used by the openstack-ansible project.

CHAPTER

SIX

TAGS

This role supports two tags: `cinder-install` and `cinder-config`

The `cinder-install` tag can be used to install and upgrade.

The `cinder-config` tag can be used to maintain configuration of the service.