
OpenStack-Ansible Documentation:

os_ceilometer role

Release 18.1.0.dev234

OpenStack-Ansible Contributors

Mar 28, 2024

CONTENTS

1 Table of Contents 3

1.1 Configuring the Telemetry (ceilometer) service 3

1.2 Default variables 4

1.3 Example playbook 13

1.4 Dependencies 13

1.5 Tags 13

This Ansible role installs and configures OpenStack ceilometer.

Meter and notification storage is configured to use a MongoDB backend by default. This role does not install and configure the MongoDB backend. Deployers wishing to use MongoDB must install and configure it prior to using this role and override `ceilometer_connection_string` variable with MongoDB connection details.

TABLE OF CONTENTS

1.1 Configuring the Telemetry (ceilometer) service

The Telemetry module (ceilometer) performs the following functions:

- Efficiently polls metering data related to OpenStack services.
- Collects event and metering data by monitoring notifications sent from services.
- Publishes collected data to various targets including data stores and message queues.

Note: As of Liberty, the alarming functionality is in a separate component. The metering-alarm containers handle the functionality through aodh services. For configuring these services, see the aodh docs: <https://docs.openstack.org/aodh/latest/>

1.1.1 Configuring the hosts

Configure ceilometer by specifying the `metering-compute_hosts` and `metering-infra_hosts` directives in the `/etc/openstack_deploy/conf.d/ceilometer.yml` file: [ceilometer.yml.example](#)

The `metering-compute_hosts` host the `ceilometer-agent-compute` service. It runs on each compute node and polls for resource utilization statistics. The `metering-infra_hosts` host several services:

- A central agent (`ceilometer-agent-central`): Runs on a central management server to poll for resource utilization statistics for resources not tied to instances or compute nodes. Multiple agents can be started to enable workload partitioning (See HA section below).
- A notification agent (`ceilometer-agent-notification`): Runs on a central management server(s) and consumes messages from the message queue(s) to build event and metering data, then sends it to one or more publishers. Multiple notification agents can be started to enable workload partitioning (See HA section below).
- An API server (`ceilometer-api`): Runs on one or more central management servers to provide data access from the data store.

1.1.2 Configuring the hosts for an HA deployment

Ceilometer supports running the polling and notification agents in an HA deployment.

The Tooz library provides the coordination within the groups of service instances. Tooz can be used with several backends. At the time of this writing, the following backends are supported:

- Zookeeper: Recommended solution by the Tooz project.
- Redis: Recommended solution by the Tooz project.
- Memcached: Recommended for testing.

Important: The OpenStack-Ansible project does not deploy these backends. One of the backends must exist before deploying the ceilometer service.

Achieve HA by configuring the proper directives in `ceilometer.conf` using `ceilometer_ceilometer_conf_overrides` in the `user_variables.yml` file. The [Ceilometer Admin Guide](#) details the options used in `ceilometer.conf` for HA deployment. The following is an example of `ceilometer_ceilometer_conf_overrides`:

```
ceilometer_ceilometer_conf_overrides:
  coordination:
    backend_url: "zookeeper://172.20.1.110:2181"
  notification:
    workload_partitioning: True
```

To clone or view the source code for this repository, visit the role repository for [os_ceilometer](#).

1.2 Default variables

```
## Verbosity Options
debug: False

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

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

# Set installation method.
ceilometer_install_method: "source"
```

(continues on next page)

(continued from previous page)

```

ceilometer_venv_python_executable: "{{ openstack_venv_python_executable |
↳ default('python3') }}"

## The git source/branch
ceilometer_git_repo: https://opendev.org/openstack/ceilometer
ceilometer_git_install_branch: master

## The packages to build from source
ceilometer_upper_constraints_url: "{{ requirements_git_url | default('https://
↳ releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↳ branch | default('master')) }}"
ceilometer_git_constraints:
  - "--constraint {{ ceilometer_upper_constraints_url }}"

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

# Name of the virtual env to deploy into
ceilometer_venv_tag: "{{ venv_tag | default('untagged') }}"
ceilometer_bin: "{{ _ceilometer_bin }}"

## System info
ceilometer_system_user_name: ceilometer
ceilometer_system_group_name: ceilometer
ceilometer_system_shell: /bin/false
ceilometer_system_comment: ceilometer system user
ceilometer_system_user_home: "/var/lib/{{ ceilometer_system_user_name }}"

## Oslo Messaging info

# RPC
ceilometer_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default(
↳ 'rabbitmq_all') }}"
ceilometer_oslomsg_rpc_setup_host: "{{ (ceilometer_oslomsg_rpc_host_group in
↳ groups) | ternary(groups[ceilometer_oslomsg_rpc_host_group][0], 'localhost
↳ ') }}"
ceilometer_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit
↳ ') }}"
ceilometer_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
ceilometer_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
ceilometer_oslomsg_rpc_userid: ceilometer
ceilometer_oslomsg_rpc_vhost: /ceilometer
ceilometer_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
↳ 'TLSv1_2') }}"
ceilometer_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('')
↳ }}"

# Notify
ceilometer_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↳ 'rabbitmq_all') }}"

```

(continues on next page)

(continued from previous page)

```

ceilometer_oslomsg_notify_setup_host: "{{ (ceilometer_oslomsg_notify_host_
↪group in groups) | ternary(groups[ceilometer_oslomsg_notify_host_group][0],
↪'localhost') }}"
ceilometer_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
ceilometer_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
ceilometer_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl |
↪default(False) }}"
ceilometer_oslomsg_notify_userid: "{{ ceilometer_oslomsg_rpc_userid }}"
ceilometer_oslomsg_notify_password: "{{ ceilometer_oslomsg_rpc_password }}"
ceilometer_oslomsg_notify_vhost: "{{ ceilometer_oslomsg_rpc_vhost }}"
ceilometer_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
↪'TLSv1_2') }}"
ceilometer_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('
↪') }}"

# Cinder notifications
cinder_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
cinder_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
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 | default('cinder
↪') }}"
cinder_oslomsg_notify_password: "{{ cinder_ceilometer_enabled |
↪ternary(cinder_oslomsg_rpc_password, '') }}"
cinder_oslomsg_notify_vhost: "{{ cinder_oslomsg_rpc_vhost | default('/cinder
↪') }}"
cinder_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
↪'TLSv1_2') }}"
cinder_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }
↪}"

# Designate notifications
designate_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
designate_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
designate_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
designate_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False),
↪}}"
designate_oslomsg_notify_userid: "{{ designate_oslomsg_rpc_userid | default(
↪'designate-rpc') }}"
designate_oslomsg_notify_password: "{{ designate_ceilometer_enabled |
↪ternary(designate_oslomsg_rpc_password, '') }}"
designate_oslomsg_notify_vhost: "{{ designate_oslomsg_rpc_vhost | default('/
↪designate') }}"
designate_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
↪'TLSv1_2') }}"

```

(continues on next page)

(continued from previous page)

```

designate_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Glance notifications
glance_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default('rabbitmq_all') }}"
glance_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default('rabbit') }}"
glance_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
glance_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
glance_oslomsg_notify_userid: "{{ glance_oslomsg_rpc_userid | default('glance') }}"
glance_oslomsg_notify_password: "{{ glance_ceilometer_enabled | ternary(glance_oslomsg_rpc_password, '') }}"
glance_oslomsg_notify_vhost: "{{ glance_oslomsg_rpc_vhost | default('/glance') }}"
glance_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_2') }}"
glance_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Heat notifications
heat_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default('rabbitmq_all') }}"
heat_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default('rabbit') }}"
heat_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
heat_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
heat_oslomsg_notify_userid: "{{ heat_oslomsg_rpc_userid | default('heat') }}"
heat_oslomsg_notify_password: "{{ heat_ceilometer_enabled | ternary(heat_oslomsg_rpc_password, '') }}"
heat_oslomsg_notify_vhost: "{{ heat_oslomsg_rpc_vhost | default('/heat') }}"
heat_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_2') }}"
heat_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Keystone notifications
keystone_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default('rabbitmq_all') }}"
keystone_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default('rabbit') }}"
keystone_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
keystone_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
keystone_oslomsg_notify_userid: "{{ keystone_oslomsg_rpc_userid | default('keystone') }}"
keystone_oslomsg_notify_password: "{{ keystone_ceilometer_enabled | ternary(keystone_oslomsg_rpc_password, '') }}"
keystone_oslomsg_notify_vhost: "{{ keystone_oslomsg_rpc_vhost | default('/keystone') }}"

```

(continues on next page)

(continued from previous page)

```

keystone_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
    ↪ 'TLSv1_2') }}"
keystone_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('
    ↪ ') }}"

# Neutron notifications
neutron_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
    ↪ 'rabbitmq_all') }}"
neutron_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
    ↪ 'rabbit') }}"
neutron_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
neutron_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
    ↪ ""
neutron_oslomsg_notify_userid: "{{ neutron_oslomsg_rpc_userid | default(
    ↪ 'neutron') }}"
neutron_oslomsg_notify_password: "{{ neutron_ceilometer_enabled | ternary(
    ↪ neutron_oslomsg_rpc_password, '') }}"
neutron_oslomsg_notify_vhost: "{{ neutron_oslomsg_rpc_vhost | default('/
    ↪ neutron') }}"
neutron_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default(
    ↪ 'TLSv1_2') }}"
neutron_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') |
    ↪ }}"

# Nova notifications
nova_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
    ↪ 'rabbitmq_all') }}"
nova_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default('rabbit
    ↪ ') }}"
nova_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
nova_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
nova_oslomsg_notify_userid: "{{ nova_oslomsg_rpc_userid | default('nova') }}"
nova_oslomsg_notify_password: "{{ nova_ceilometer_enabled | ternary(nova_
    ↪ oslomsg_rpc_password, '') }}"
nova_oslomsg_notify_vhost: "{{ nova_oslomsg_rpc_vhost | default('/nova') }}"
nova_oslomsg_notify_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_
    ↪ 2') }}"
nova_oslomsg_notify_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Octavia notifications
octavia_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
    ↪ 'rabbitmq_all') }}"
octavia_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
    ↪ 'rabbit') }}"
octavia_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
octavia_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
    ↪ ""
octavia_oslomsg_notify_userid: "{{ octavia_oslomsg_rpc_userid | default(
    ↪ 'octavia') }}"

```

(continues on next page)

(continued from previous page)

```

octavia_oslmsg_notify_password: "{{ octavia_ceilometer_enabled |
↳ ternary(octavia_oslmsg_rpc_password, '') }}"
octavia_oslmsg_notify_vhost: "{{ octavia_oslmsg_rpc_vhost | default('/'
↳ octavia') }}"
octavia_oslmsg_notify_ssl_version: "{{ oslmsg_rpc_ssl_version | default(
↳ 'TLSv1_2') }}"
octavia_oslmsg_notify_ssl_ca_file: "{{ oslmsg_rpc_ssl_ca_file | default('')
↳ }}"

# Sahara notifications
sahara_oslmsg_notify_host_group: "{{ oslmsg_notify_host_group | default(
↳ 'rabbitmq_all') }}"
sahara_oslmsg_notify_transport: "{{ oslmsg_notify_transport | default(
↳ 'rabbit') }}"
sahara_oslmsg_notify_port: "{{ oslmsg_notify_port | default('5672') }}"
sahara_oslmsg_notify_use_ssl: "{{ oslmsg_notify_use_ssl | default(False) }}"
sahara_oslmsg_notify_userid: "{{ sahara_oslmsg_rpc_userid | default('sahara
↳ ') }}"
sahara_oslmsg_notify_password: "{{ sahara_ceilometer_enabled |
↳ ternary(sahara_oslmsg_rpc_password, '') }}"
sahara_oslmsg_notify_vhost: "{{ sahara_oslmsg_rpc_vhost | default('/sahara
↳ ') }}"
sahara_oslmsg_notify_ssl_version: "{{ oslmsg_rpc_ssl_version | default(
↳ 'TLSv1_2') }}"
sahara_oslmsg_notify_ssl_ca_file: "{{ oslmsg_rpc_ssl_ca_file | default('')
↳ }}"

# Swift notifications
swift_oslmsg_notify_host_group: "{{ oslmsg_notify_host_group | default(
↳ 'rabbitmq_all') }}"
swift_oslmsg_notify_transport: "{{ oslmsg_notify_transport | default('rabbit
↳ ') }}"
swift_oslmsg_notify_port: "{{ oslmsg_notify_port | default('5672') }}"
swift_oslmsg_notify_use_ssl: "{{ oslmsg_notify_use_ssl | default(False) }}"
swift_oslmsg_notify_userid: "swift"
swift_oslmsg_notify_vhost: "/swift"
swift_oslmsg_notify_ssl_version: "{{ oslmsg_rpc_ssl_version | default(
↳ 'TLSv1_2') }}"
swift_oslmsg_notify_ssl_ca_file: "{{ oslmsg_rpc_ssl_ca_file | default('')
↳ }}"

# Trove notifications
trove_oslmsg_notify_host_group: "{{ oslmsg_notify_host_group | default(
↳ 'rabbitmq_all') }}"
trove_oslmsg_notify_transport: "{{ oslmsg_notify_transport | default('rabbit
↳ ') }}"
trove_oslmsg_notify_port: "{{ oslmsg_notify_port | default('5672') }}"
trove_oslmsg_notify_use_ssl: "{{ oslmsg_notify_use_ssl | default(False) }}"
trove_oslmsg_notify_userid: "{{ trove_oslmsg_rpc_userid | default('trove')
↳ }}"

```

(continues on next page)

(continued from previous page)

```

trove_oslmsg_notify_password: "{{ trove_ceilometer_enabled | ternary(trove_
↳oslmsg_rpc_password, '') }}"
trove_oslmsg_notify_vhost: "{{ trove_oslmsg_rpc_vhost | default('/trove') }}"
↳
trove_oslmsg_notify_ssl_version: "{{ oslmsg_rpc_ssl_version | default(
↳'TLSv1_2') }}"
trove_oslmsg_notify_ssl_ca_file: "{{ oslmsg_rpc_ssl_ca_file | default('') }}"
↳

## (Qdrouterd) info
# TODO(ansmith): Change structure when more backends will be supported
ceilometer_oslmsg_amqp1_enabled: "{{ ceilometer_oslmsg_rpc_transport ==
↳'amqp' }}"

#Ceilometer services info
ceilometer_role_name: admin

## Service Type and Data
ceilometer_service_region: "{{ service_region | default('RegionOne') }}"
ceilometer_service_user_name: ceilometer
ceilometer_service_project_name: service
ceilometer_service_project_domain_id: default
ceilometer_service_user_domain_id: default
ceilometer_service_interface: internalURL

ceilometer_service_in_ldap: "{{ service_ldap_backend_enabled | default(False)
↳ }}"

## Ceilometer config
# Cap the maximum number of threads / workers when a user value is unspecified
ceilometer_notification_workers_max: 16
ceilometer_notification_workers: "{{ [[ansible_facts['processor_vcpus
↳']|default(2) // 2, 1] | max, ceilometer_notification_workers_max] | min }}"

# Enable/Disable Ceilometer per service
cinder_ceilometer_enabled: "{{ (groups['cinder_all'] is defined) and (groups[
↳'cinder_all'] | length > 0) }}"
designate_ceilometer_enabled: "{{ (groups['designate_all'] is defined) and
↳(groups['designate_all'] | length > 0) }}"
glance_ceilometer_enabled: "{{ (groups['glance_all'] is defined) and (groups[
↳'glance_all'] | length > 0) }}"
heat_ceilometer_enabled: "{{ (groups['heat_all'] is defined) and (groups[
↳'heat_all'] | length > 0) }}"
keystone_ceilometer_enabled: "{{ (groups['keystone_all'] is defined) and
↳(groups['keystone_all'] | length > 0) }}"
neutron_ceilometer_enabled: "{{ (groups['neutron_all'] is defined) and
↳(groups['neutron_all'] | length > 0) }}"
nova_ceilometer_enabled: "{{ (groups['nova_all'] is defined) and (groups[
↳'nova_all'] | length > 0) }}"

```

(continues on next page)

(continued from previous page)

```

octavia_ceilometer_enabled: "{{ (groups['octavia_all'] is defined) and_
↳(groups['octavia_all'] | length > 0) }}"
sahara_ceilometer_enabled: "{{ (groups['sahara_all'] is defined) and (groups[
↳'sahara_all'] | length > 0) }}"
swift_ceilometer_enabled: "{{ (groups['swift_all'] is defined) and (groups[
↳'swift_all'] | length > 0) }}"
trove_ceilometer_enabled: "{{ (groups['trove_all'] is defined) and (groups[
↳'trove_all'] | length > 0) }}"

# Enable/Disable Telemetry partner services in Ceilometer
# Defaults to True if the service is deployed.
ceilometer_aodh_enabled: "{{ groups['aodh_all'] is defined and groups['aodh_
↳all'] | length > 0 }}"
ceilometer_gnocchi_enabled: "{{ groups['gnocchi_all'] is defined and groups[
↳'gnocchi_all'] | length > 0 }}"

# TODO (andymccr): Remove in Queen release (along with config template_
↳overrides)
# Sampling interval for ceilometer-polling services
ceilometer_sample_interval: 600
ceilometer_meter_sample_interval: "{{ ceilometer_sample_interval }}"
ceilometer_cpu_source_sample_interval: "{{ ceilometer_sample_interval }}"
ceilometer_disk_source_sample_interval: "{{ ceilometer_sample_interval }}"
ceilometer_network_source_sample_interval: "{{ ceilometer_sample_interval }}"

## Keystone authentication middleware
ceilometer_keystone_auth_plugin: "{{ ceilometer_keystone_auth_type }}"
ceilometer_keystone_auth_type: password

# Common pip packages
ceilometer_pip_packages:
  - "git+{{ ceilometer_git_repo }}@{{ ceilometer_git_install_branch }}"
  ↳#egg=ceilometer"
  - ceilometermiddleware
  - cryptography
  - gnocchiclient
  - osprofiler
  - PyMySQL
  - pymongo
  - python-memcached
  - systemd-python
  - tooz[zake]
  - warlock
  - kazoo
  - redis

ceilometer_optional_oslmsg_amqp1_pip_packages:
  - oslo.messaging[amqp1]

```

(continues on next page)

(continued from previous page)

```

ceilometer_agent_init_overrides: {}
ceilometer_polling_init_overrides: {}

## Service Name-Group Mapping
ceilometer_services:
  ceilometer-agent-notification:
    group:
      - ceilometer_agent_notification
    config_overrides: "{{ ceilometer_agent_init_overrides }}"
    service_name: ceilometer-agent-notification
    execstarts: "{{ ceilometer_bin }}/ceilometer-agent-notification"
  ceilometer-polling:
    group:
      - ceilometer_agent_central
      - ceilometer_agent_compute
    config_overrides: "{{ ceilometer_polling_init_overrides }}"
    service_name: ceilometer-polling
    execstarts: "{{ ceilometer_bin }}/ceilometer-polling"

ceilometer_required_secrets:
  - memcached_encryption_key
  - ceilometer_telemetry_secret
  - ceilometer_oslmsg_rpc_password
  - ceilometer_oslmsg_notify_password
  - ceilometer_service_password

#: Tunable file-based overrides
# The contents of these files, if they exist, are read from the
# specified path on the deployment host, interpreted by the
# template engine and copied to the target host. If they do
# not exist then the default files will be sourced from the
# service git repository.
ceilometer_data_meters_default_file_path: "/etc/openstack_deploy/ceilometer/
↪meters.yaml"
ceilometer_event_definitions_default_file_path: "/etc/openstack_deploy/
↪ceilometer/event_definitions.yaml"
ceilometer_event_pipeline_default_file_path: "/etc/openstack_deploy/
↪ceilometer/event_pipeline.yaml"
ceilometer_gnocchi_resources_default_file_path: "/etc/openstack_deploy/
↪ceilometer/gnocchi_resources.yaml"
ceilometer_loadbalancer_v2_meter_definitions_default_file_path: "/etc/
↪openstack_deploy/ceilometer/loadbalancer_v2_meter_definitions.yaml"
ceilometer_osprofiler_event_definitions_default_file_path: "/etc/openstack_
↪deploy/ceilometer/osprofiler_event_definitions.yaml"
ceilometer_polling_default_file_path: "/etc/openstack_deploy/ceilometer/
↪polling.yaml"
ceilometer_pipeline_default_file_path: "/etc/openstack_deploy/ceilometer/
↪pipeline.yaml"

```

(continues on next page)

(continued from previous page)

```
## Tunable var-based overrides
# The contents of these are templated over the default files.
ceilometer_ceilometer_conf_overrides: {}
ceilometer_data_meters_yaml_overrides: {}
ceilometer_event_definitions_yaml_overrides: {}
ceilometer_event_pipeline_yaml_overrides: {}
ceilometer_gnocchi_resources_yaml_overrides: {}
ceilometer_loadbalancer_v2_meter_definitions_yaml_overrides: {}
ceilometer_osprofiler_event_definitions_yaml_overrides: {}
ceilometer_pipeline_yaml_overrides: {}
ceilometer_polling_yaml_overrides: {}
ceilometer_rootwrap_conf_overrides: {}
```

1.3 Example playbook

```
---
- name: Install ceilometer server
  hosts: ceilometer_all
  user: root
  roles:
    - { role: "os_ceilometer", tags: [ "ceilometer-config", "ceilometer-
↪install" ] }
  vars:
    external_lb_vip_address: 172.16.24.1
    internal_lb_vip_address: 192.168.0.1
```

1.4 Dependencies

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

1.5 Tags

This role supports two tags: `ceilometer-install` and `ceilometer-config`. The `ceilometer-install` tag can be used to install and upgrade. The `ceilometer-config` tag can be used to maintain the configuration of the service.