
OpenStack-Ansible Documentation:

os_trove role

Release 18.1.0.dev269

OpenStack-Ansible Contributors

Mar 23, 2026

CONTENTS

1	Configuring Trove	3
1.1	Setup a neutron network for use by trove	3
1.2	Building Trove images	4
1.3	Use stand-alone RabbitMQ	5
2	Default variables	7
3	Dependencies	19
4	Example playbook	21

This Ansible role installs Trove.

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

CONFIGURING TROVE

Note

Care should be taken when deploying Trove in production environments. Be sure to fully understand the security implications of the deployed architecture.

Trove provides DBaaS to an OpenStack deployment. It deploys guest VMs that provide the desired DB for use by the end consumer. The trove guest VMs need connectivity back to the trove services via RPC (oslo.messaging) and the OpenStack services. The way these guest VM get access to those services could be via internal networking (in the case of oslo.messaging) or via public interfaces (in the case of OpenStack services). For the example configuration, we'll designate a provider network as the network for trove to provision on each guest VM. The guest can then connect to oslo.messaging via this network and to the OpenStack services externally. Optionally, the guest VMs could use the internal network to access OpenStack services, but that would require more containers being bound to this network.

The deployment configuration outlined below may not be appropriate for production environments. Review this very carefully with your own security requirements.

1.1 Setup a neutron network for use by trove

Trove needs connectivity between the control plane and the DB guest VMs. For this purpose a provider network should be created which bridges the trove containers (if the control plane is installed in a container) or hosts with VMs. In a general case, neutron networking can be a simple flat network. An example entry into `openstack_user_config.yml` is shown below:

```
- network:
  container_bridge: "br-dbaas"
  container_type: "veth"
  container_interface: "eth13"
  host_bind_override: "eth13"
  ip_from_q: "dbaas"
  type: "flat"
  net_name: "dbaas-mgmt"
  group_binds:
    - neutron_linuxbridge_agent
    - oslomsg_rpc
    - trove_all
```

Make sure to modify the other entries in this file as well.

The `net_name` will be the physical network that is specified when creating the neutron network. The default value of `dbaas-mgmt` is also used to lookup the addresses of the rpc messaging container. If the default is not used then some variables in `defaults/main.yml` will need to be overwritten.

By default this role will not create the neutron network automatically. However, the default values can be changed to create the neutron network. See the `trove_service_net_*` variable in `defaults/main.yml`. By customizing the `trove_service_net_*` variables and having this role create the neutron network a full deployment of the OpenStack and DBaaS can proceed without interruption or intervention.

The following is an example how to set up a provider network in neutron manually, if so desired:

```
neutron net-create dbaas_service_net --shared \
    --provider:network_type flat \
    --provider:physical_network dbaas-mgmt

neutron subnet-create dbaas_service_net 172.29.252.0/22 --name dbaas_service_
↪ subnet
    --ip-version=4 \
    --allocation-pool start=172.29.252.110,end=172.29.255.
↪ 255 \
    --enable-dhcp \
    --dns-nameservers list=true 8.8.4.4 8.8.8.8
```

Special attention needs to be applied to the `--allocation-pool` to not have ips which overlap with ips assigned to hosts or containers (see the `used_ips` variable in `openstack_user_config.yml`)

Note

This role needs the neutron network created before it can run properly since the trove guest agent configuration file contains that information.

1.2 Building Trove images

When building disk image for the guest VM deployments there are many items to consider. Listed below are a few:

1. Security of the VM and the network infrastructure
2. What DBs will be installed
3. What DB services will be supported
4. How will the images be maintained

Images can be built using the `diskimage-builder` tooling. The trove virtual environment can be tar'd up from the trove containers and deployed to the images using custom `diskimage-builder` elements.

See the `trove/integration/scripts/files/elements` directory contents in the OpenStack Trove project for `diskimage-builder` elements to build trove disk images.

1.3 Use stand-alone RabbitMQ

Since Trove uses RabbitMQ to interact with guest servers it requires you to pass the neutron network into the RabbitMQ container which is a security risk. As a result, you might want to isolate Trove from other services in terms of the RabbitMQ cluster and use a standalone one.

In order to deploy new RabbitMQ cluster and use it for Trove, you will need to:

1. Create a new group for RabbitMQ containers. You will need to create a file inside `/etc/openstack_deploy/env.d` which defines group mappings

```
component_skel:
  trove_rabbitmq:
    belongs_to:
      - trove_mq_all

container_skel:
  trove_rabbit_container:
    belongs_to:
      - trove-mq_containers
    contains:
      - trove_rabbitmq

physical_skel:
  trove-mq_containers:
    belongs_to:
      - all_containers
  trove-mq_hosts:
    belongs_to:
      - hosts
```

2. Define on which hosts this group will be deployed. This can be done either with a new file in `conf.d` or inside `openstack_user_config.yml`

```
trove-mq_hosts:
  aio1:
    ip: 172.29.236.100
```

3. Add to the dbaas network mapping for the new group:

```
- network:
  container_bridge: "br-dbaas"
  container_type: "veth"
  container_interface: "eth14"
  host_bind_override: "eth14"
  ip_from_q: "dbaas"
  type: "flat"
  net_name: "dbaas-mgmt"
  group_binds:
    - neutron_linuxbridge_agent
    - oslmsg_rpc
    - trove_rabbitmq
```

4. Create overrides for dedicated rabbitmq containers, ie /etc/openstack_deploy/group_vars/trove_rabbitmq.yml

```
rabbitmq_cluster_name: trove
rabbitmq_cookie_token: <token>
rabbitmq_monitoring_password: <password>
```

5. Create overrides for trove service containers, ie /etc/openstack_deploy/group_vars/trove_all.yml

Note

For notifications we still want to use main RabbitMQ cluster

```
trove_guest_rpc_host_group: trove_rabbitmq
trove_guest_oslmsg_rpc_password: SecretPassword
```

6. Run playbooks to create rabbitmq containers and deploy cluster on them

```
openstack-ansible playbooks/lxc-containers-create.yml --limit_
↪trove_rabbitmq,lxc_hosts
openstack-ansible playbooks/rabbitmq-install.yml -e rabbitmq_
↪host_group=trove_rabbitmq
```

DEFAULT VARIABLES

```
#
# (c) 2016 Donovan Francesco <donovan.francesco@is.co.za>
# (c) 2016 Paul Stevens <paul.stevens@is.co.za>

# python venv executable
trove_venv_python_executable: "{{ openstack_venv_python_executable | default(
  ↪'python3') }}"

# Set the host which will execute the shade modules
# for the service setup. The host must already have
# clouds.yaml properly configured.
trove_service_setup_host: "{{ openstack_service_setup_host | default(
  ↪'localhost') }}"
trove_service_setup_host_python_interpreter: >-
  {{
    openstack_service_setup_host_python_interpreter | default(
      (trove_service_setup_host == 'localhost') | ternary(ansible_playbook_
  ↪python, ansible_facts['python']['executable']))
  }}

trove_package_state: "{{ package_state | default('latest') }}"

debug: false
trove_system_group_name: trove
trove_system_user_name: trove
trove_system_user_comment: Trove System User
trove_system_user_shell: /bin/false
trove_system_user_home: "/var/lib/{{ trove_system_user_name }}"
trove_log_directory: /var/log/trove
trove_etc_directory: /etc/trove

trove_service_name: trove
trove_service_user_name: trove
trove_service_type: database
trove_service_description: "OpenStack DBaaS (Trove)"
trove_service_project_name: service
trove_service_admin_role_names:
  - admin
```

(continues on next page)

(continued from previous page)

```

- service
trove_service_token_roles:
- service
trove_service_token_roles_required: "{{ openstack_service_token_roles_
↳required | default(True) }}"
trove_service_region: "{{ service_region | default('RegionOne') }}"
trove_service_endpoint_type: internal
trove_service_host: "{{ openstack_service_bind_address | default('0.0.0.0') }}
↳"
trove_service_port: 8779
trove_service_proto: http
trove_service_publicuri_proto: "{{ openstack_service_publicuri_proto |
↳default(trove_service_proto) }}"
trove_service_internaluri_proto: "{{ openstack_service_internaluri_proto |
↳default(trove_service_proto) }}"
trove_service_adminuri_proto: "{{ openstack_service_adminuri_proto |
↳default(trove_service_proto) }}"
trove_service_publicurl: "{{ trove_service_publicuri_proto }}://{{ external_
↳lb_vip_address }}:{{ trove_service_port }}/v1.0/(tenant_id)s"
trove_service_internalurl: "{{ trove_service_internaluri_proto }}://{{
↳internal_lb_vip_address }}:{{ trove_service_port }}/v1.0/(tenant_id)s"
trove_service_adminurl: "{{ trove_service_adminuri_proto }}://{{ internal_lb_
↳vip_address }}:{{ trove_service_port }}/v1.0/(tenant_id)s"
trove_auth_url: "{{ keystone_service_internalurl }}"

trove_service_in_ldap: "{{ service_ldap_backend_enabled | default(False) }}"

trove_profiler_enabled: false

## Cap the maximum number of threads / workers when a user value is
↳unspecified.
trove_api_workers_max: 16
trove_api_workers: >-
  {{ [(ansible_facts['processor_vcpus'] // ansible_facts['processor_threads_
↳per_core']) | default(1), 1] | max * 2, trove_api_workers_max] | min }}

# uWSGI settings
trove_wsgi_threads: 1
trove_use_uwsgi: true
trove_uwsgi_tls:
  crt: "{{ trove_ssl_cert }}"
  key: "{{ trove_ssl_key }}"

## Cap the maximum number of threads / workers when a user value is
↳unspecified.
trove_conductor_workers_max: 16
trove_conductor_workers: >-
  {{ [(ansible_facts['processor_vcpus'] // ansible_facts['processor_threads_
↳per_core']) | default(1), 1] | max * 2, trove_conductor_workers_max] | min }}

```

(continues on next page)

(continued from previous page)

```

↪}

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

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

# Name of the virtual env to deploy into
trove_venv_tag: "{{ venv_tag | default('untagged') }}"
trove_bin: "/openstack/venvs/trove-{{ trove_venv_tag }}/bin"

trove_git_repo: "https://opendev.org/openstack/trove"
trove_git_install_branch: master
trove_upper_constraints_url: >-
  {{ requirements_git_url | default('https://releases.openstack.org/
↪constraints/upper/' ~ requirements_git_install_branch | default('master')) }
↪}
trove_git_constraints:
  - "--constraint {{ trove_upper_constraints_url }}"

# Database vars
trove_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
trove_db_setup_python_interpreter: >-
  {{
    openstack_db_setup_python_interpreter | default(
      (trove_db_setup_host == 'localhost') | ternary(ansible_playbook_python,
↪ansible_facts['python']['executable'])
  }}
trove_galera_address: "{{ galera_address | default('127.0.0.1') }}"
trove_galera_database_name: trove
trove_galera_user: trove
trove_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
trove_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('') }}"
trove_galera_port: "{{ galera_port | default('3306') }}"
trove_galera_connection_string: >-
  mysql+pymysql://{{ trove_galera_user }}:{{ trove_galera_password }}@{{ trove_
↪galera_address }}:{{ trove_galera_port }}/{{ trove_galera_database_name
  }}?charset=utf8{% if trove_galera_use_ssl | bool %}&ssl_verify_cert=true{%
  if trove_galera_ssl_ca_cert | length > 0 %}&ssl_ca={{ trove_galera_ssl_
↪ca_cert }}{% endif %}{% endif %}
trove_db_max_overflow: "{{ openstack_db_max_overflow | default('50') }}"
trove_db_max_pool_size: "{{ openstack_db_max_pool_size | default('5') }}"
trove_db_pool_timeout: "{{ openstack_db_pool_timeout | default('30') }}"
trove_db_connection_recycle_time: "{{ openstack_db_connection_recycle_time |
↪default('600') }}"

## Oslo Messaging vars
# RPC

```

(continues on next page)

(continued from previous page)

```

trove_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default('rabbitmq_
↳all') }}"
trove_oslomsg_rpc_setup_host: "{{ (trove_oslomsg_rpc_host_group in groups) | t
↳ternary(groups[trove_oslomsg_rpc_host_group][0], 'localhost') }}"
trove_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit') }}"
trove_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1') }}"
trove_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
trove_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
trove_oslomsg_rpc_userid: trove
trove_oslomsg_rpc_policies: []
# vhost name depends on value of oslomsg_rabbit_quorum_queues. In case quorum_
↳queues
# are not used - vhost name will be prefixed with leading `/.
trove_oslomsg_rpc_vhost:
  - name: /trove
    state: "{{ trove_oslomsg_rabbit_quorum_queues | ternary('absent', 'present
↳') }}"
  - name: trove
    state: "{{ trove_oslomsg_rabbit_quorum_queues | ternary('present', 'absent
↳') }}"
trove_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_2
↳') }}"
trove_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Notify
trove_oslomsg_notify_configure: "{{ oslomsg_notify_configure | default(trove_
↳ceilometer_enabled) }}"
trove_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↳'rabbitmq_all') }}"
trove_oslomsg_notify_setup_host: "{{ (trove_oslomsg_notify_host_group in_
↳groups) | ternary(groups[trove_oslomsg_notify_host_group][0], 'localhost') }
↳}"
trove_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default('rabbit
↳') }}"
trove_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.0.1
↳') }}"
trove_oslomsg_notify_port: "{{ oslomsg_notify_port | default('5672') }}"
trove_oslomsg_notify_use_ssl: "{{ oslomsg_notify_use_ssl | default(False) }}"
trove_oslomsg_notify_userid: "{{ trove_oslomsg_rpc_userid }}"
trove_oslomsg_notify_password: "{{ trove_oslomsg_rpc_password }}"
trove_oslomsg_notify_vhost: "{{ trove_oslomsg_rpc_vhost }}"
trove_oslomsg_notify_ssl_version: "{{ oslomsg_notify_ssl_version | default(
↳'TLSv1_2') }}"
trove_oslomsg_notify_ssl_ca_file: "{{ oslomsg_notify_ssl_ca_file | default('
↳') }}"
trove_oslomsg_notify_policies: []

## RabbitMQ integration
trove_oslomsg_rabbit_quorum_queues: "{{ oslomsg_rabbit_quorum_queues |_

```

(continues on next page)

(continued from previous page)

```

↪default(True) }}"
trove_oslomsg_rabbit_stream_fanout: "{{ oslomsg_rabbit_stream_fanout |
↪default(trove_oslomsg_rabbit_quorum_queues) }}"
trove_oslomsg_rabbit_transient_quorum_queues: "{{ oslomsg_rabbit_transient_
↪quorum_queues | default(trove_oslomsg_rabbit_stream_fanout) }}"
trove_oslomsg_rabbit_qos_prefetch_count: "{{ oslomsg_rabbit_qos_prefetch_
↪count | default(trove_oslomsg_rabbit_stream_fanout | ternary(10, 0)) }}"
trove_oslomsg_rabbit_queue_manager: "{{ oslomsg_rabbit_queue_manager |
↪default(trove_oslomsg_rabbit_quorum_queues) }}"
trove_oslomsg_rabbit_quorum_delivery_limit: "{{ oslomsg_rabbit_quorum_
↪delivery_limit | default(0) }}"
trove_oslomsg_rabbit_quorum_max_memory_bytes: "{{ oslomsg_rabbit_quorum_max_
↪memory_bytes | default(0) }}"
trove_guest_oslomsg_rabbit_quorum_queues: "{{ oslomsg_rabbit_quorum_queues |
↪default(True) }}"
trove_guest_oslomsg_rabbit_stream_fanout: "{{ oslomsg_rabbit_stream_fanout |
↪default(trove_guest_oslomsg_rabbit_quorum_queues) }}"
trove_guest_oslomsg_rabbit_transient_quorum_queues: "{{ oslomsg_rabbit_
↪transient_quorum_queues | default(trove_guest_oslomsg_rabbit_stream_fanout)
↪}}"
trove_guest_oslomsg_rabbit_qos_prefetch_count: "{{ oslomsg_rabbit_qos_
↪prefetch_count | default(trove_guest_oslomsg_rabbit_stream_fanout |
↪ternary(10, 0)) }}"
trove_guest_oslomsg_rabbit_queue_manager: "{{ oslomsg_rabbit_queue_manager |
↪default(trove_guest_oslomsg_rabbit_quorum_queues) }}"

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

# Rabbit vars
trove_control_exchange: trove
trove_rabbit_notification_topic: notification

# The trove guest agent in the deployed DB VMs need access to OpenStack
↪services (keystone, swift, etc)
# and also to rabbitmq. The way it gets access (networking) these services
↪may differ.
# By default assume:
#   OpenStack services are accesses through public interfaces
#   Infrastructure services (rabbitmq) are accessed through the defined
↪provider network.
#
# The value of 'net_name' field of the provider network network to use for
↪infrastructure services
trove_provider_net_name: dbaas-mgmt
trove_provider_network: >-
  {{
    provider_networks | map(attribute='network') | selectattr('net_name',
↪'defined') | selectattr(

```

(continues on next page)

(continued from previous page)

```

        'net_name', 'equalto', trove_provider_net_name
    ) | list | first
}}
# The name of the network interface
trove_provider_net_iface: "{{ (is_metal | bool) | ternary(trove_provider_
↪network['container_bridge'], trove_provider_network['container_interface'])_
↪ }}"
trove_guest_endpoint_type: public

# Guestagent RPC configuration
trove_guest_rpc_host_group: "{{ oslmsg_rpc_host_group | default('rabbitmq_all
↪') }}"
trove_guest_oslmsg_rpc_setup_host: "{{ (trove_guest_rpc_host_group in_
↪groups) | ternary(groups[trove_guest_rpc_host_group][0], 'localhost') }}"
trove_guest_oslmsg_rpc_servers: >-
    {{
        groups[trove_guest_rpc_host_group] | map('extract', hostvars, 'ansible_
↪facts') | map(attribute=trove_provider_net_iface | replace('-', '_')
        ) | map(attribute='ipv4.address') | join(',')
    }}
trove_guest_oslmsg_rpc_use_ssl: "{{ trove_oslmsg_rpc_use_ssl }}"
trove_guest_oslmsg_rpc_port: "{{ trove_oslmsg_rpc_port }}"
trove_guest_oslmsg_rpc_userid: "{{ trove_oslmsg_rpc_userid }}"
trove_guest_oslmsg_rpc_password: "{{ trove_oslmsg_rpc_password }}"
trove_guest_oslmsg_rpc_ssl_version: "{{ trove_oslmsg_rpc_ssl_version }}"
trove_guest_oslmsg_rpc_ssl_ca_file: "{{ trove_oslmsg_rpc_ssl_ca_file }}"
trove_guest_oslmsg_rpc_policies: []
trove_guest_oslmsg_rpc_vhost:
  - name: /trove
    state: "{{ trove_guest_oslmsg_rabbit_quorum_queues | ternary('absent',
↪'present') }}"
  - name: trove
    state: "{{ trove_guest_oslmsg_rabbit_quorum_queues | ternary('present',
↪'absent') }}"

trove_guest_oslmsg_notify_configure: "{{ oslmsg_notify_configure |_
↪default(trove_ceilometer_enabled) }}"
trove_guest_notify_host_group: "{{ oslmsg_notify_host_group | default(
↪'rabbitmq_all') }}"
trove_guest_oslmsg_notify_servers: >-
    {{
        groups[trove_guest_notify_host_group] | map('extract', hostvars, 'ansible_
↪facts') | map(attribute=trove_provider_net_iface | replace('-', '_')
        ) | map(attribute='ipv4.address') | join(',')
    }}
trove_guest_oslmsg_notify_use_ssl: "{{ trove_oslmsg_notify_use_ssl }}"
trove_guest_oslmsg_notify_port: "{{ trove_oslmsg_notify_port }}"
trove_guest_oslmsg_notify_userid: "{{ trove_oslmsg_notify_userid }}"
trove_guest_oslmsg_notify_password: "{{ trove_oslmsg_notify_password }}"

```

(continues on next page)

(continued from previous page)

```

trove_guest_oslmsg_notify_vhost: "{{ trove_guest_oslmsg_rpc_vhost }}"
trove_guest_oslmsg_notify_ssl_version: "{{ trove_oslmsg_notify_ssl_version }}
↪}"
trove_guest_oslmsg_notify_ssl_ca_file: "{{ trove_oslmsg_notify_ssl_ca_file }}
↪}"
trove_guest_oslmsg_notify_policies: []

# Trove image settings.
# Set the directory where the downloaded images will be stored
# on the trove_service_setup_host host. If the host is localhost,
# then the user running the playbook must have access to it.
trove_image_local_path: "{{ lookup('env', 'HOME') }}/openstack-ansible/trove"
trove_image_path_owner: "{{ lookup('env', 'USER') }}"
## Example Glance Image - Fedora Atomic
# - name: ubuntu-bionic          #Name of the image in Glance
#   disk_format: qcow2          #Disk format (e.g. qcow2)
#   image_format: bare          #Image format
#   public: true                #Boolean - is the image public
#   file: https://tarballs.opendev.org/openstack/trove/images/trove-master-
↪guest-ubuntu-bionic.qcow2
#   tags:
#     - trove
#   checksum:
↪"sha256:9a5252e24b82a5edb1ce75b05653f59895685b0f1028112462e908a12daae518"
trove_guestagent_images: []

# For OpenStack services that have public, admin, and internal access, use
↪the public ones for the guest VMs.
trove_guest_auth_url: "{{ keystone_service_publicurl }}"
trove_guest_swift_url: "{{ trove_service_publicuri_proto }}://{{ external_lb_
↪vip_address }}:{{ swift_proxy_port | default(8080) }}/v1/AUTH_"

trove_resources_deploy_host: localhost
trove_resources_deploy_python_interpreter: "{{ ansible_playbook_python }}"
trove_guest_ssh_enabled: false
trove_guest_ssh_key_manage: true
trove_guest_ssh_key_name: trove_guestagent_key
trove_guest_ssh_key_dir: "{{ openstack_ssh_keypairs_dir | default(lookup('env
↪', 'HOME') ~ '/.ssh') }}"
trove_guest_ssh_key_comment: "Generated-By-OpenStack-Ansible"
# Options: ssh, pkcs1 and pkcs8
trove_guest_ssh_key_format: ssh
# Options: rsa, dsa, rsa1, ecdsa, ed25519
trove_guest_ssh_key_type: rsa
trove_guest_ssh_key_size: 4096
# Security group which will allow SSH connections
trove_guest_ssh_security_group_name: trove_guest_management
trove_guest_ssh_security_group_extra_rules: []

```

(continues on next page)

(continued from previous page)

```

trove_swift_enabled: >-
  {{
    (groups['swift_all'] is defined and groups['swift_all'] | length > 0) or
    (groups['ceph-rgw'] is defined and groups['ceph-rgw'] | length > 0) or
    (ceph_rgws is defined and ceph_rgws | length > 0)
  }}
trove_designate_enabled: "{{ (groups['designate_all'] is defined and groups[
↪ 'designate_all'] | length > 0) }}"
trove_cinder_enabled: "{{ (groups['cinder_volume'] is defined and groups[
↪ 'cinder_volume'] | length > 0) }}"

trove_service_neutron_endpoint_type: "{{ trove_service_endpoint_type }}"
trove_service_cinder_endpoint_type: "{{ trove_service_endpoint_type }}"
trove_service_nova_endpoint_type: "{{ trove_service_endpoint_type }}"
trove_service_glance_endpoint_type: "{{ trove_service_endpoint_type }}"
trove_service_swift_endpoint_type: "{{ trove_service_endpoint_type }}"

trove_dns_domain_name: "trove.com."
trove_dns_domain_id: "00000000-0000-0000-0000-000000000000"
# Notification topics for designate.
trove_notifications_designate: notifications_designate

# Trove service network settings.
# These values are used when creating an OpenStack network to be used by ↵
↪ Trove. By default the network will
# not be created.
trove_service_net_setup: false
trove_service_net_validate_certs: "False"
trove_service_net_phys_net: dbaas-mgmt
trove_service_net_type: flat
trove_service_net_name: dbaas_service_net
# Network segmentation ID if vlan, gre...
# trove_service_net_segmentation_id:
trove_service_subnet_name: dbaas_subnet
trove_service_net_subnet_cidr: "172.29.252.0/22"
trove_service_net_dhcp: "True"
trove_service_net_allocation_pool_start: "172.29.252.110"
trove_service_net_allocation_pool_end: "172.29.255.254"
trove_service_net_endpoint_type: "{{ trove_service_endpoint_type }}"

# UUID of security groups that will be attached to the management net of ↵
↪ guests
trove_management_security_groups: []

# A set of datastores which will be populated and maintained for the ↵
↪ deployment.
# trove_datastores:
#   mariadb: # name of the manager
#     - name: MariaDB # name of the datastore

```

(continues on next page)

(continued from previous page)

```

#     versions:
#         - name: 11.4
#           enabled: true
#           version-number: 11.4.7
#           default: true
#         - name: 11.4
#           enabled: false
#           version-number: 11.4.6

trove_datastores: {}

# RPC encryption keys
# See the Trove documentation as to the significance of the rpc encryption_
# keys
# Trove supplies default values but we enforce they not be left to their_
# default values
trove_enable_secure_rpc_messaging: "True"
trove_required_secrets:
  - trove_galera_password
  - trove_oslmsg_rpc_password
  - trove_oslmsg_notify_password
  - trove_service_password
  - trove_admin_user_password
  - trove_taskmanager_rpc_encr_key
  - trove_inst_rpc_key_encr_key
  - trove_instance_rpc_encr_key

# Keystone AuthToken/Middleware
trove_keystone_auth_plugin: "{{ trove_keystone_auth_type }}"
trove_keystone_auth_type: password
trove_service_project_domain_name: Default
trove_service_user_domain_name: Default
trove_service_project_domain_id: default
trove_service_user_domain_id: default

# Glance images
trove_glance_images: []

trove_pip_packages:
  - cryptography
  - os-client-config
  - osprofiler
  - pexpect
  - PyMySQL
  - pymemcache
  - python-troveclient
  - python-memcached
  - systemd-python
  - "git+{{ trove_git_repo }}@{{ trove_git_install_branch }}#egg=trove"

```

(continues on next page)

(continued from previous page)

```
# Memcached override
trove_memcached_servers: "{{ memcached_servers }}"

## Tunable overrides
trove_config_overrides: {}
trove_api_paste_ini_overrides: {}
trove_guestagent_config_overrides: {}
trove_policy_overrides: {}
trove_api_init_config_overrides: {}
trove_api_uwsgi_ini_overrides: {}
trove_conductor_init_config_overrides: {}
trove_taskmanager_init_config_overrides: {}

## Service Name-Group Mapping
trove_services:
  trove-api:
    group: trove_api
    service_name: trove-api
    execstarts: "{{ trove_bin }}/trove-api"
    init_config_overrides: "{{ trove_api_init_config_overrides }}"
    wsgi_app: "{{ trove_use_uwsgi }}"
    wsgi_name: trove-api-wsgi
    uwsgi_overrides: "{{ trove_api_uwsgi_ini_overrides }}"
    uwsgi_bind_address: "{{ trove_service_host }}"
    uwsgi_port: "{{ trove_service_port }}"
    uwsgi_tls: "{{ trove_backend_ssl | ternary(trove_uwsgi_tls, {}) }}"
    start_order: 1
  trove-conductor:
    group: trove_conductor
    service_name: trove-conductor
    execstarts: "{{ trove_bin }}/trove-conductor"
    init_config_overrides: "{{ trove_conductor_init_config_overrides }}"
    start_order: 2
  trove-taskmanager:
    group: trove_taskmanager
    service_name: trove-taskmanager
    execstarts: "{{ trove_bin }}/trove-taskmanager"
    init_config_overrides: "{{ trove_taskmanager_init_config_overrides }}"
    start_order: 3

###
### Backend TLS
###

# Define if communication between haproxy and service backends should be
# encrypted with TLS.
trove_backend_ssl: "{{ openstack_service_backend_ssl | default(False) }}"
```

(continues on next page)

(continued from previous page)

```

# Storage location for SSL certificate authority
trove_pki_dir: "{{ openstack_pki_dir | default('/etc/openstack_deploy/pki') }}"
↪"

# Delegated host for operating the certificate authority
trove_pki_setup_host: "{{ openstack_pki_setup_host | default('localhost') }}"

# trove server certificate
trove_pki_keys_path: "{{ trove_pki_dir ~ '/certs/private/' }}"
trove_pki_certs_path: "{{ trove_pki_dir ~ '/certs/certs/' }}"
trove_pki_intermediate_cert_name: "{{ openstack_pki_service_intermediate_cert_
↪name | default('ExampleCorpIntermediate') }}"
trove_pki_regen_cert: ""
trove_pki_san: "{{ openstack_pki_san | default({'dns': [ansible_facts[
↪'hostname']], 'ip': [management_address]}) }}"
trove_pki_backend: "{{ openstack_pki_backend | default('standalone') }}"
trove_pki_certificates:
- name: "trove_{{ ansible_facts['hostname'] }}"
  cn: "{{ ansible_facts['hostname'] }}"
  san: "{{ trove_pki_san }}"
  # standalone backend only
  provider: ownca
  signed_by: "{{ trove_pki_intermediate_cert_name }}"

# trove destination files for SSL certificates
trove_ssl_cert: /etc/trove/trove.pem
trove_ssl_key: /etc/trove/trove.key

# Installation details for SSL certificates
trove_pki_install_certificates:
- name: "trove_{{ ansible_facts['hostname'] }}"
  type: "certificate_chain"
  dest: "{{ trove_ssl_cert }}"
  owner: "{{ trove_system_user_name }}"
  group: "{{ trove_system_user_name }}"
  # standalone backend only
  src: "{{ trove_user_ssl_cert | default(trove_pki_certs_path ~ 'trove_' ~
↪ansible_facts['hostname'] ~ '-chain.crt') }}"
- name: "trove_{{ ansible_facts['hostname'] }}"
  type: "private_key"
  dest: "{{ trove_ssl_key }}"
  owner: "{{ trove_system_user_name }}"
  group: "{{ trove_system_user_name }}"
  # standalone backend only
  src: "{{ trove_user_ssl_key | default(trove_pki_keys_path ~ 'trove_' ~
↪ansible_facts['hostname'] ~ '.key.pem') }}"

# Define user-provided SSL certificates
trove_user_ssl_cert: ""
trove_user_ssl_key: ""

```


DEPENDENCIES

To use this role, define the following variables:

```
# trove TCP listening port
trove_service_port: 8779

# Service and user passwords
trove_galera_password:
trove_rabbitmq_password:
trove_service_password:
trove_admin_user_password:

# Trove RPC encryption keys.
trove_taskmanager_rpc_encr_key:
trove_inst_rpc_key_encr_key:
trove_instance_rpc_encr_key:
```

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

EXAMPLE PLAYBOOK

```
---  
- name: Install trove  
  hosts: all  
  user: root  
  roles:  
    - role: "os_trove"
```