OpenStack-Ansible Documentation: galera_server role

Release 18.1.0.dev256

OpenStack-Ansible Contributors

Apr 13, 2023

CONTENTS

Ansible role to install and configure a Galera cluster powered by MariaDB

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

CHAPTER

ONE

DEFAULT VARIABLES

```yaml
# Set the package install state for distribution packages
# Options are 'present' and 'latest'
galera_package_state: "latest"

galera_cluster_members: "{{ groups['galera_all'] }}"
galera_server_bootstrap_node: "{{ galera_cluster_members[0] }}"
galera_ignore_cluster_state: false
galera_upgrade: false
galera_force_bootstrap: false

galera_wsrep_node_name: "{{ inventory_hostname }}"
galera_cluster_name: openstack_galera_cluster
galera_server_bind_address: "{{ openstack_service_bind_address | default('0.0.
↪0.0') }}"

# The galera server-id should be set on all cluster nodes to ensure
#  that replication is handled correctly and the error
#  "Warning: You should set server-id to a non-0 value if master_host is
#   set; we will force server id to 2, but this MySQL server will not act
#   as a slave." is no longer present.
# galera_server_id: 0

# These are here to stub out the internal ROLE API.
#  if these are used they should be set within the
#  distro specific variable files found in vars/
galera_debconf_items: []
galera_mariadb_service_name: mariadb
galera_mariadb_server_package: "{{ _galera_mariadb_server_package }}"

# The major version used to select the repo URL path
galera_major_version: 10.6
galera_minor_version: 9

# Set the URL for the MariaDB repository
galera_repo_host: "downloads.mariadb.com"
galera_repo_url: "{{ _galera_repo_url }}"

# Set the repo information for the MariaDB repository
```

(continues on next page)

(continued from previous page)

```
galera_repo: "{{ _galera_repo }}"

# Set the gpg keys needed to be imported
# This should be a list of dicts, with each dict
# giving a set of arguments to the applicable
# package module. The following is an example for
# systems using the apt package manager.
# galera_gpg_keys:
#   - id: '0xF1656F24C74CD1D8'
#     keyserver: 'hkp://keyserver.ubuntu.com:80'
#     validate_certs: no
galera_gpg_keys: "{{ _galera_gpg_keys | default([]) }}"

galera_monitoring_user: monitoring
galera_monitoring_user_password: ""

# WARNING: Set this to open xinetd rules for galera monitoring.
# This is REQUIRED to run a working openstack-ansible deployment.
# If it's undefined the galera cluster state can't be reported,
# and haproxy would fail to do proper load balancing on the cluster.
# Because this opens connections to the cluster status, this
# should be restricted, which we do in the integrated build.
# Please override accordingly to your use case.
# This can be replaced with other hostnames, cidr, ips, and ips + wildcards.
#
#galera_monitoring_allowed_source: "0.0.0.0/0"

# Enable or disable the installation of galera development packages
galera_install_devel: false

# Enable or disable the installation of galera server
galera_install_server: false

# Enable or disable the galera monitoring check capability
galera_monitoring_check_enabled: true

# Set the monitoring port used with the galera monitoring check.
galera_monitoring_check_port: 9200

galera_root_user: root

# WARNING: This option is deprecated and will be removed in v12.0
galera_gcache_size: 1024M

galera_max_heap_table_size: 32M
galera_tmp_table_size: 32M
galera_tmp_dir: /var/lib/mysql/#tmp
galera_ignore_db_dirs:
  - "'#tmp'"
```

(continues on next page)

(continued from previous page)

```
galera_file_limits: 65535
galera_wait_timeout: "{{ openstack_db_connection_recycle_time | default('600
↪') }}"
# Increase this value if large SST transfers cause mysql startup to fail due
# to timeout
galera_startup_timeout: 1800

## innodb options
galera_innodb_buffer_pool_size: 4096M
galera_innodb_log_file_size: 1024M
galera_innodb_log_buffer_size: 128M

## wsrep configuration
galera_wsrep_address: "{{ ansible_host }}"
galera_wsrep_address_port: "{{ galera_wsrep_address }}:3306"
galera_wsrep_cluster_port: 4567
galera_wsrep_cluster_address: >-
  {% set _var = [] -%}
  {% for cluster_host in galera_cluster_members -%}
  {% set _addr = hostvars[cluster_host]['galera_wsrep_address']
                 | default(hostvars[cluster_host]['ansible_host']) -%}
  {% if _var.append(_addr) %}{% endif -%}
  {% endfor -%}
  {# If only 1 cluster member is present output an empty string so the
     single-node member will re-bootstrap correctly upon restart #}
  {{ _var | join(',') if galera_cluster_members | length > 1 else '' }}

galera_wsrep_node_incoming_address: "{{ galera_wsrep_address }}"
## Cap the maximum number of threads / workers when a user value is␣
↪unspecified.
galera_wsrep_slave_threads_max: 16
galera_wsrep_slave_threads: "{{ [[ansible_facts['processor_vcpus']|default(2),
↪ 2] | max, galera_wsrep_slave_threads_max] | min }}"
galera_wsrep_retry_autocommit: 3
galera_wsrep_debug: NONE
galera_wsrep_sst_method: mariabackup
galera_wsrep_provider_options:
  - { option: "gcache.size", value: "{{ galera_gcache_size }}" }
  - { option: "gmcast.listen_addr", value: "tcp://{{ galera_wsrep_node_
↪incoming_address }}:{{ galera_wsrep_cluster_port  }}" }
galera_wsrep_sst_auth_user: "{{ galera_root_user }}"
galera_wsrep_sst_auth_password: "{{ galera_root_password  }}"

# mariabackup parallel/sync threads
galera_mariabackup_threads: 4

# Galera slow/unindexed query logging
galera_slow_query_logging: 0
```

(continues on next page)

(continued from previous page)

```
galera_unindexed_query_logging: 0

## Tunable overrides
galera_my_cnf_overrides: {}
galera_cluster_cnf_overrides: {}
galera_debian_cnf_overrides: {}
galera_encryption_overrides: {}
galera_init_overrides: {}

# Set the max connections value for galera. Set this value to override the
# computed value which is (100 x vCPUs) with a cap of 1600. If computed, the
# lowest value throughout the cluster will be used which is something to note
# if deploying galera on different hardware.
# galera_max_connections: 500

# This is only applied if the ansible_facts['pkg_mgr'] is 'apt'
galera_distro_package_pins:
  - package: '*'
    release: MariaDB
    priority: 1001

# Galera Server SSL functionality.

# Storage location for SSL certificate authority
galera_pki_dir: "{{ openstack_pki_dir | default('/etc/pki/galera-ca') }}"

# Create a certificate authority if one does not already exist
galera_pki_create_ca: "{{ openstack_pki_authorities is not defined | bool }}"
galera_pki_regen_ca: ''

galera_pki_authorities:
  - name: "MariaDBRoot"
    country: "GB"
    state_or_province_name: "England"
    organization_name: "Example Corporation"
    organizational_unit_name: "IT Security"
    cn: "MariaDB Root CA"
    provider: selfsigned
    basic_constraints: "CA:TRUE"
    key_usage:
      - digitalSignature
      - cRLSign
      - keyCertSign
    not_after: "+3650d"
  - name: "MariaDBIntermediate"
    country: "GB"
    state_or_province_name: "England"
    organization_name: "Example Corporation"
    organizational_unit_name: "IT Security"
```

(continues on next page)

(continued from previous page)

```
    cn: "MariaDB Intermediate CA"
    provider: ownca
    basic_constraints: "CA:TRUE,pathlen:0"
    key_usage:
      - digitalSignature
      - cRLSign
      - keyCertSign
    not_after: "+3650d"
    signed_by: "MariaDBRoot"

# Installation details for certificate authorities
galera_pki_install_ca:
  - name: "MariaDBRoot"
    condition: "{{ galera_pki_create_ca }}"

# Galera server certificate
galera_pki_keys_path: "{{ galera_pki_dir ~ '/certs/private/' }}"
galera_pki_certs_path: "{{ galera_pki_dir ~ '/certs/certs/' }}"
galera_pki_intermediate_cert_name: "{{ openstack_pki_service_intermediate_
↪cert_name | default('MariaDBIntermediate') }}"
galera_pki_intermediate_cert_path: "{{ galera_pki_dir ~ '/roots/' ~ galera_
↪pki_intermediate_cert_name ~ '/certs/' ~ galera_pki_intermediate_cert_name ~
↪ '.crt' }}"
galera_pki_regen_cert: ''
galera_pki_certificates:
  - name: "galera_{{ ansible_facts['hostname'] }}"
    provider: ownca
    cn: "{{ ansible_facts['hostname'] }}"
    san: "{{ 'DNS:' ~ ansible_facts['hostname'] ~ ',' ~ ((galera_address |␣
↪ansible.netcommon.ipaddr) is string) | ternary('IP', 'DNS') ~ ':' ~ galera_
↪address }}"
    signed_by: "{{ galera_pki_intermediate_cert_name }}"

galera_use_ssl: false
galera_ssl_verify: true
galera_ssl_cert: /etc/ssl/certs/galera.pem
galera_ssl_key: /etc/mysql/ssl/galera.key
galera_ssl_ca_cert: /etc/ssl/certs/galera-ca.pem

## These options should be specified in user_variables if necessary,␣
↪otherwise self-signed certs are used.
# galera_user_ssl_cert: /etc/openstack_deploy/self_signed_certs/galera.pem
# galera_user_ssl_key: /etc/openstack_deploy/self_signed_certs/galera.key
# galera_user_ssl_ca_cert: /etc/openstack_deploy/self_signed_certs/galera-ca.
↪pem

# This option is used for creating the CA and overriding the Galera address␣
↪on the clients side.
# Should be set to either internal VIP or VIP FQDN, depending on what is␣
↪currently used in the env.
```

(continues on next page)

(continued from previous page)

```yaml
galera_address: "{{ ansible_host }}"

# Installation details for SSL certificates
galera_pki_install_certificates:
  - src: "{{ galera_user_ssl_cert | default(galera_pki_certs_path ~ 'galera_'␣
↪~ ansible_facts['hostname'] ~ '-chain.crt') }}"
    dest: "{{ galera_ssl_cert }}"
    owner: "root"
    group: "root"
    mode: "0644"
  - src: "{{ galera_user_ssl_key | default(galera_pki_keys_path ~ 'galera_' ~␣
↪ansible_facts['hostname'] ~ '.key.pem') }}"
    dest: "{{ galera_ssl_key }}"
    owner: "mysql"
    group: "root"
    mode: "0600"
  - src: "{{ galera_user_ssl_ca_cert | default(galera_pki_intermediate_cert_
↪path) }}"
    dest: "{{ galera_ssl_ca_cert }}"
    owner: "root"
    group: "root"
    mode: "0644"


# MariaDB 10.1+ ships with 'PrivateDevices=True' in the systemd unit file. This
# provides some additional security, but it causes problems with systemd 219.
# While the security enhancements are helpful on bare metal hosts with␣
↪multiple
# services running, they are not as helpful when MariaDB is running in a
# container with its own isolated namespaces.
#
# Related bugs:
#   https://bugs.launchpad.net/openstack-ansible/+bug/1697531
#   https://github.com/lxc/lxc/issues/1623
#   https://github.com/systemd/systemd/issues/6121
#
# Setting the following variable to 'yes' will disable the PrivateDevices
galera_disable_privatedevices: "{{ _galera_disable_privatedevices }}"

#install and configure the galera client as well as the server
galera_install_client: false
galera_client_package_install: "{{ galera_install_client }}"
galera_client_package_state: "latest"
galera_client_drop_config_file: "true"
galera_client_my_cnf_overrides: {}

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

(continues on next page)

(continued from previous page)

```
## Database info
galera_db_setup_host: "{{ openstack_db_setup_host | default(galera_cluster_
↪members[0] | default('localhost')) }}"
galera_db_setup_python_interpreter: "{{ openstack_db_setup_python_interpreter␣
↪| default((galera_db_setup_host == 'localhost') | ternary(ansible_playbook_
↪python, ansible_facts['python']['executable'])) }}"

# Configure backups of database
# copies is the number of full backups to be kept, the corresponding
# incremental backups will also be kept. Uses systemd timer instead of cron.
galera_mariadb_backups_enabled: false
#galera_mariadb_backups_group_gid: <specify a GID>
galera_mariadb_backups_group_name: backups
galera_mariadb_backups_path: "/var/backup/mariadb_backups"
galera_mariadb_backups_full_copies: 2
galera_mariadb_backups_full_on_calendar: "*-*-* 00:00:00"
galera_mariadb_backups_increment_on_calendar:
  - "*-*-* 06:00:00"
  - "*-*-* 12:00:00"
  - "*-*-* 18:00:00"
#galera_mariadb_backups_user is the name of the mariadb database user
galera_mariadb_backups_user: galera_mariadb_backup
galera_mariadb_backups_suffix: "{{ inventory_hostname }}"
galera_mariadb_backups_cnf_file: "/etc/mysql/mariabackup.cnf"
galera_mariadb_backups_nodes: ["{{ galera_cluster_members[0] }}"]

galera_mariadb_encryption_enabled: false
galera_mariadb_encryption_plugin: "file_key_management"
galera_db_encryption_tmp_dir: ""
```

CHAPTER

TWO

REQUIRED VARIABLES

To use this role, define the following variables:

```
galera_root_password: secrete
```

CHAPTER

THREE

EXAMPLE PLAYBOOK

```
---
- name: Install Galera server
  hosts: galera_all
  user: root
  roles:
    - galera_server
  vars:
    galera_install_server: true
    galera_install_client: true
    galera_root_password: secrete
```

CHAPTER

FOUR

EXTERNAL RESTART HOOKS

When the role performs a restart of the mariadb 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`, that way external roles can manage the load balancer endpoints responsible for sending traffic to the MariaDB 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.