
OpenStack-Ansible Documentation: **os_gnocchi role** *Release 18.1.0.dev181*

OpenStack-Ansible Contributors

May 02, 2023

CONTENTS

1	Default variables	3
2	Example playbook	9
3	Dependencies	11
4	Tags	13

This Ansible role installs and configures OpenStack gnocchi.

The gnocchi API is served using Apache mod_wsgi by default.

This role supports configuration of file, swift and ceph storage backends. By default, the file backend is used.

This role also ships with an Ansible library, *gnocchi* that the role uses to manage archive policies and archive policy rules. By default, three policies are configured: low, medium, and high. A single archive policy rule is configured setting the *low* policy as the default for all metrics.

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

DEFAULT VARIABLES

```
#: Special role execution lifecycles
# Only create Gnocchi's identity entities in Keystone
gnocchi_identity_only: False

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

#venv python version
gnocchi_venv_python_executable: "{{ openstack_venv_python_executable |
↳ default('python3') }}"

#: Enable for debug logging level
debug: false

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

# Toggle keystone authentication for gnocchi
gnocchi_auth_mode: "keystone"

# These variables are used in 'developer mode' in order to allow the role
# to build an environment directly from a git source without the presence
# of an OpenStack-Ansible repo_server.
gnocchi_git_repo: https://github.com/gnocchixyz/gnocchi
gnocchi_git_install_branch: master
gnocchi_upper_constraints_url: "{{ requirements_git_url | default('https://
↳ releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↳ branch | default('master')) }}"
gnocchi_git_constraints:
- "--constraint {{ gnocchi_upper_constraints_url }}"
```

(continues on next page)

(continued from previous page)

```

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

#: Use of deprecated config options will cause a fatal application error
gnocchi_fatal_deprecations: false

#: External SSL forwarding proto, assumes TLS termination at load balancer
gnocchi_ssl_external: "{{ openstack_external_ssl | default(True) }}"
gnocchi_secure_proxy_ssl_header: HTTP_X_FORWARDED_PROTO

#: Name of the virtual env to deploy into
gnocchi_venv_tag: "{{ venv_tag | default('untagged') }}"
gnocchi_bin: "/openstack/venvs/gnocchi-{{ gnocchi_venv_tag }}/bin"
gnocchi_venv_pkgs: "/openstack/venvs/gnocchi-{{ gnocchi_venv_tag }}/lib/
↳python2.7/site-packages"

#: Set the etc dir path where gnocchi is installed.
# This is used for role access to the db migrations.
# Example:
# gnocchi_etc_dir: "/usr/local/etc/gnocchi"
gnocchi_etc_dir: "{{ gnocchi_bin | dirname }}/etc/gnocchi"

#: Index Database info
gnocchi_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
gnocchi_db_setup_python_interpreter: "{{ openstack_db_setup_python_
↳interpreter | default((gnocchi_db_setup_host == 'localhost') |
↳ternary(ansible_playbook_python, ansible_facts['python']['executable'])) }}"
gnocchi_galera_address: "{{ galera_address | default('127.0.0.1') }}"
gnocchi_galera_database: gnocchi
gnocchi_galera_user: gnocchi
gnocchi_db_sync_options: ""
gnocchi_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
gnocchi_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('') }}"
gnocchi_galera_port: "{{ galera_port | default('3306') }}"

#: Storage info
gnocchi_storage_driver: file
gnocchi_storage_file_basepath: "{{ gnocchi_system_user_home }}"
gnocchi_storage_swift_container_prefix: "gnocchi"
gnocchi_storage_redis_url: "redis://localhost:6379/"

gnocchi_coordination_url: "mysql://{{ gnocchi_galera_user }}:{{ gnocchi_
↳container_mysql_password }}@{{ gnocchi_galera_address }}/{{ gnocchi_galera_
↳database }}?charset=utf8&timeout=5{% if gnocchi_galera_use_ssl | bool %}&
↳ssl_ca={{ gnocchi_galera_ssl_ca_cert }}{% endif %}"

# Incoming configuration
# Incoming configuration is not applied if driver is the same as storage one
gnocchi_incoming_driver: "{{ gnocchi_storage_driver }}"

```

(continues on next page)

(continued from previous page)

```

gnocchi_incoming_file_basepath: "{{ gnocchi_storage_file_basepath }}"
gnocchi_incoming_swift_container_prefix: "{{ gnocchi_storage_swift_container_
↳prefix }}"
gnocchi_incoming_redis_url: "{{ gnocchi_storage_redis_url }}"

#: Default Ceph parameters
gnocchi_ceph_pool: "metrics"
gnocchi_ceph_username: "gnocchi"
# If gnocchi_storage_driver == gnocchi_incoming_driver this would have no_
↳effect
gnocchi_ceph_incoming_pool: "{{ gnocchi_ceph_pool }}"
gnocchi_ceph_incoming_username: "{{ gnocchi_ceph_username }}"

#a System info
gnocchi_system_user_name: gnocchi
gnocchi_system_group_name: gnocchi
gnocchi_system_shell: /bin/false
gnocchi_system_comment: gnocchi system user
gnocchi_system_user_home: "/var/lib/{{ gnocchi_system_user_name }}"

#: Service Type and Data
gnocchi_service_name: gnocchi
gnocchi_service_type: metric
gnocchi_service_description: "OpenStack Metric Service"
gnocchi_service_project_description: "OpenStack Services"
gnocchi_keystone_auth_plugin: "{{ gnocchi_keystone_auth_type }}"
gnocchi_keystone_auth_type: password
gnocchi_service_region: "{{ service_region | default('RegionOne') }}"
gnocchi_service_user_name: gnocchi
gnocchi_service_role_names:
- admin
- service
gnocchi_service_token_roles:
- service
gnocchi_service_token_roles_required: "{{ openstack_service_token_roles_
↳required | default(True) }}"
gnocchi_service_project_name: service
gnocchi_service_project_domain_id: default
gnocchi_service_user_domain_id: default
gnocchi_service_address: "{{ openstack_service_bind_address | default('0.0.0.0
↳') }}"
gnocchi_service_port: 8041
gnocchi_service_proto: http
gnocchi_service_registry_proto: "{{ gnocchi_service_proto }}"
gnocchi_service_publicuri_proto: "{{ openstack_service_publicuri_proto |_
↳default(gnocchi_service_proto) }}"
gnocchi_service_adminuri_proto: "{{ openstack_service_adminuri_proto |_
↳default(gnocchi_service_proto) }}"
gnocchi_service_internaluri_proto: "{{ openstack_service_internaluri_proto |_

```

(continues on next page)

(continued from previous page)

```

↪default(gnocchi_service_proto) }}"
gnocchi_service_publicuri: "{{ gnocchi_service_publicuri_proto }}://{{↪
↪external_lb_vip_address }}:{{ gnocchi_service_port }}"
gnocchi_service_publicurl: "{{ gnocchi_service_publicuri }}"
gnocchi_service_internaluri: "{{ gnocchi_service_internaluri_proto }}://{{↪
↪internal_lb_vip_address }}:{{ gnocchi_service_port }}"
gnocchi_service_internalurl: "{{ gnocchi_service_internaluri }}"
gnocchi_service_adminuri: "{{ gnocchi_service_adminuri_proto }}://{{ internal_↪
↪lb_vip_address }}:{{ gnocchi_service_port }}"
gnocchi_service_adminurl: "{{ gnocchi_service_adminuri }}"

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

## uWSGI setup
gnocchi_wsgi_threads: 1
gnocchi_wsgi_processes_max: 16
gnocchi_wsgi_processes: "{{ [(ansible_facts['processor_vcpus']//ansible_↪
↪facts['processor_threads_per_core'])|default(1), 1] | max * 2, gnocchi_wsgi_↪
↪processes_max] | min }}"

gnocchi_metricd_workers_max: 16
gnocchi_metricd_workers: "{{ [(ansible_facts['processor_vcpus']//ansible_↪
↪facts['processor_threads_per_core'])|default(1), 1] | max * 2, gnocchi_↪
↪metricd_workers_max] | min }}"

gnocchi_uwsgi_conf_overrides: {}
gnocchi_api_init_overrides: {}
gnocchi_metricd_init_overrides: {}

## Service Names
gnocchi_services:
  gnocchi-api:
    group: "gnocchi_api"
    service_name: "gnocchi-api"
    service_enabled: true
    init_config_overrides: "{{ gnocchi_api_init_overrides }}"
    wsgi_app: True
    wsgi_path: "{{ gnocchi_bin }}/gnocchi-api"
    uwsgi_bind_address: "{{ gnocchi_service_address }}"
    uwsgi_port: "{{ gnocchi_service_port }}"
    uwsgi_overrides: "{{ gnocchi_uwsgi_conf_overrides }}"
  gnocchi-metricd:
    group: "gnocchi_metricd"
    service_name: "gnocchi-metricd"
    service_enabled: true
    init_config_overrides: "{{ gnocchi_metricd_init_overrides }}"
    execstarts: "{{ gnocchi_bin }}/gnocchi-metricd"

gnocchi_pip_package_extras:

```

(continues on next page)

(continued from previous page)

```

- "{{ (gnocchi_auth_mode == 'keystone') | ternary('keystone', '') }}"
- mysql
- "{{ gnocchi_storage_driver_pip_extra }}"
- "{{ gnocchi_incoming_driver_pip_extra }}"

#: Common pip packages
gnocchi_pip_packages:
- cryptography
- "git+{{ gnocchi_git_repo }}@{{ gnocchi_git_install_branch }}#egg=gnocchi[{{
↪{ gnocchi_pip_package_extras | unique | reject('equalto', '') | join(',') }}
↪]"
- gnocchiclient
- osprofiler
- pymemcache
- python-memcached
- kazoo
- systemd-python

# Memcached override
gnocchi_memcached_servers: "{{ memcached_servers }}"

#: 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.
gnocchi_api_paste_default_file_path: "/etc/openstack_deploy/gnocchi/api-paste.
↪ini"
gnocchi_policy_default_file_path: "/etc/openstack_deploy/gnocchi/policy.json"

# If the above-mentioned files do not exist, then these
# paths will be used to find the files from the git config
# lookup location.
gnocchi_git_config_lookup_location: https://raw.githubusercontent.com/
↪gnocchixyz/gnocchi/{{ gnocchi_git_install_branch }}/
gnocchi_api_paste_git_file_path: "gnocchi/rest/api-paste.ini"
gnocchi_policy_git_file_path: "gnocchi/rest/policy.json"

#: Tunable var-based overrides
# The contents of these are templated over the default files.
gnocchi_api_paste_ini_overrides: {}
gnocchi_conf_overrides: {}
gnocchi_policy_overrides: {}

```


EXAMPLE PLAYBOOK

```
- name: Installation and setup of Gnocchi
hosts: gnocchi_all
user: root
roles:
  - role: "os_gnocchi"
    tags:
      - "os-gnocchi"
    galera_root_user: root
vars_prompt:
  - name: "galera_root_password"
    prompt: "What is galera_root_password?"
```


DEPENDENCIES

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

TAGS

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