OpenStack-Ansible Documentation: os_glance role

Release 18.1.0.dev282

OpenStack-Ansible Contributors

Mar 28, 2024

CONTENTS

1 Configuring the Image (glance) service **1**

1.1 Configuring default and additional stores 1

1.2 Storing images in Cloud Files 2

1.3 Special considerations 3

2 Default variables **5**

3 Dependencies **13**

4 Example playbook **15**

5 External Restart Hooks **17**

6 Tags **19**

CHAPTER

ONE

CONFIGURING THE IMAGE (GLANCE) SERVICE

In an all-in-one deployment with a single infrastructure node, the Image (glance) service uses the local file system on the target host to store images. When deploying production clouds, we recommend backing glance with a swift backend or some form of shared storage.

1.1 Configuring default and additional stores

OpenStack-Ansible provides two configurations for controlling where glance stores files: the default store and additional stores. glance stores images in file-based storage by default. Two additional stores, `http` and `cinder` (Block Storage), are also enabled by default.

You can choose alternative default stores and alternative additional stores. For example, a deployer that uses Ceph can configure the following Ansible variables:

```
glance_default_store = rbd
glance_additional_stores:
  - swift
  - http
  - cinder
```

The configuration above configures glance to use `rbd` (Ceph) by default, but `glance_additional_stores` list enables `swift`, `http` and `cinder` stores in the glance configuration files.

The following example sets glance to use the `images` pool. This example uses cephx authentication and requires an existing `glance` account for the `images` pool.

In `user_variables.yml`:

```
glance_default_store: rbd
ceph_mons:
  - 172.29.244.151
  - 172.29.244.152
  - 172.29.244.153
```

You can use the following variables if you are not using the defaults:

```
glance_ceph_client: <glance-username>
glance_rbd_store_pool: <glance-pool-name>
glance_rbd_store_chunk_size: <chunk-size>
```

1.2 Storing images in Cloud Files

The following procedure describes how to modify the `/etc/openstack_deploy/user_variables.yml` file to enable Cloud Files usage.

1. Change the default store to use Object Storage (swift), the underlying architecture of Cloud Files:

```
glance_default_store: swift
```

2. Set the appropriate authentication URL and version:

```
glance_swift_store_auth_version: 2
glance_swift_store_auth_address: https://127.0.0.1/v2.0
```

3. Set the swift account credentials:

```
# Replace this capitalized variables with actual data.
glance_swift_store_user: GLANCE_SWIFT_TENANT:GLANCE_SWIFT_USER
glance_swift_store_key: SWIFT_PASSWORD_OR_KEY
```

4. Change the `glance_swift_store_endpoint_type` from the default `internalURL` settings to `publicURL` if needed.

```
glance_swift_store_endpoint_type: publicURL
```

5. Define the store name:

```
glance_swift_store_container: STORE_NAME
```

   Replace `STORE_NAME` with the container name in swift to be used for storing images. If the container does not exist, it is automatically created.

6. Define the store region:

```
glance_swift_store_region: STORE_REGION
```

   Replace `STORE_REGION` if needed.

7. (Optional) Set the paste deploy flavor:

```
glance_flavor: GLANCE_FLAVOR
```

   By default, glance uses caching and authenticates with the Identity (keystone) service. The default maximum size of the image cache is 10GB. The default glance container size is 12GB. In some configurations, glance attempts to cache an image which exceeds the available disk space. If necessary, you can disable caching. For example, to use Identity without caching, replace `GLANCE_FLAVOR` with `keystone`:

```
glance_flavor: keystone
```

   Or, to disable both authentication and caching, set `GLANCE_FLAVOR` to no value:

```
glance_flavor:
```

This option is set by default to use authentication and cache management in the `playbooks/roles/os_glance/defaults/main.yml` file. To override the default behavior, set `glance_flavor` to a different value in `/etc/openstack_deploy/user_variables.yml`.

The possible values for `GLANCE_FLAVOR` are:

- (Nothing)
- `caching`
- `cachemanagement`
- `keystone`
- `keystone+caching`
- `keystone+cachemanagement` (default)
- `trusted-auth`
- `trusted-auth+cachemanagement`

1.3 Special considerations

If the swift password or key contains a dollar sign (`$`), it must be escaped with an additional dollar sign (`$$`). For example, a password of `super$ecure` would need to be entered as `super$$ecure`. This is necessary due to the way oslo.config formats strings.

This role installs the following Systemd services:

- glance-api

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

CHAPTER

TWO

DEFAULT VARIABLES

```yaml
## Verbosity Options
debug: False

# Set installation method
glance_install_method: "{{ service_install_method | default('source') }}"
glance_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.
glance_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
glance_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((glance_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'
glance_package_state: "{{ package_state | default('latest') }}"

glance_git_repo: https://opendev.org/openstack/glance
glance_git_install_branch: master
glance_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
glance_git_constraints:
  - "--constraint {{ glance_upper_constraints_url }}"

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

# Name of the virtual env to deploy into
glance_venv_tag: "{{ venv_tag | default('untagged') }}"
glance_bin: "{{ _glance_bin }}"

# Set the etc dir path where glance is installed.
#  This is used for role access to the db migrations.
```

(continues on next page)

(continued from previous page)

```
#  Example:
#  glance_etc_dir: "/usr/local/etc/glance"
glance_etc_dir: "/etc/glance"

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

glance_profiler_enabled: False
glance_fatal_deprecations: False

## System info
glance_system_user_name: glance
glance_system_group_name: glance
glance_system_shell: /bin/false
glance_system_comment: glance system user
glance_system_user_home: "/var/lib/{{ glance_system_user_name }}"

## Manually specified nova UID/GID
# Deployers can specify a UID for the glance user as well as the GID for the
# glance group if needed. This is commonly used in environments where shared
# storage is used, such as NFS or GlusterFS, and glance 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.
#
# glance_system_user_uid: <UID>
# glance_system_group_gid: <GID>

glance_default_store: file
# For support of multiple backends provide `glance_additional_stores` in the␣
↪format:
# glance_additional_stores:
#   - name: private_store
#     type: http
glance_additional_stores:
  - http
  - cinder
glance_flavor: "{% if glance_default_store == 'rbd' %}keystone{% else %}
↪keystone+cachemanagement{% endif %}"
glance_show_image_direct_url: "{{ glance_default_store == 'rbd' }}"
glance_show_multiple_locations: "{{ glance_default_store == 'rbd' }}"

glance_memcached_servers: "{{ memcached_servers }}"

## API options
glance_enable_v2_api: True
```

(continues on next page)

(continued from previous page)

```
## Oslo Messaging Info

# RPC
glance_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default('rabbitmq_
↪all') }}"
glance_oslomsg_rpc_setup_host: "{{ (glance_oslomsg_rpc_host_group in groups)␣
↪| ternary(groups[glance_oslomsg_rpc_host_group][0], 'localhost') }}"
glance_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit') }}
↪"
glance_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1') }}"
glance_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
glance_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
glance_oslomsg_rpc_userid: glance
glance_oslomsg_rpc_vhost: /glance
glance_oslomsg_rpc_ssl_version: "{{ oslomsg_rpc_ssl_version | default('TLSv1_2
↪') }}"
glance_oslomsg_rpc_ssl_ca_file: "{{ oslomsg_rpc_ssl_ca_file | default('') }}"

# Notify
glance_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
glance_oslomsg_notify_setup_host: "{{ (glance_oslomsg_notify_host_group in␣
↪groups) | ternary(groups[glance_oslomsg_notify_host_group][0], 'localhost')␣
↪}}"
glance_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
glance_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.0.1
↪') }}"
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 }}"
glance_oslomsg_notify_password: "{{ glance_oslomsg_rpc_password }}"
glance_oslomsg_notify_vhost: "{{ glance_oslomsg_rpc_vhost }}"
glance_oslomsg_notify_ssl_version: "{{ oslomsg_notify_ssl_version | default(
↪'TLSv1_2') }}"
glance_oslomsg_notify_ssl_ca_file: "{{ oslomsg_notify_ssl_ca_file | default('
↪') }}"

## (Qdrouterd) integration
# TODO(ansmith): Change structure when more backends will be supported
glance_oslomsg_amqp1_enabled: "{{ glance_oslomsg_rpc_transport == 'amqp' }}"

## Database info
glance_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
glance_db_setup_python_interpreter: "{{ openstack_db_setup_python_interpreter␣
↪| default((glance_db_setup_host == 'localhost') | ternary(ansible_playbook_
↪python, ansible_facts['python']['executable'])) }}"
glance_galera_address: "{{ galera_address | default('127.0.0.1') }}"
glance_galera_database: glance
```

(continues on next page)

(continued from previous page)

```
glance_galera_user: glance
glance_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
glance_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('') }}"
glance_galera_port: "{{ galera_port | default('3306') }}"
glance_db_max_overflow: "{{ openstack_db_max_overflow | default('50') }}"
glance_db_max_pool_size: "{{ openstack_db_max_pool_size | default('5') }}"
glance_db_pool_timeout: "{{ openstack_db_pool_timeout | default('30') }}"
glance_db_connection_recycle_time: "{{ openstack_db_connection_recycle_time |␣
↪default('600') }}"

glance_role_name: admin
glance_api_bind_address: "{{ openstack_service_bind_address | default('0.0.0.0
↪') }}"
glance_api_service_port: 9292

## Service Type and Data
glance_service_region: "{{ service_region | default('RegionOne') }}"
glance_service_name: glance
glance_service_port: 9292
glance_service_proto: http
glance_service_publicuri_proto: "{{ openstack_service_publicuri_proto |␣
↪default(glance_service_proto) }}"
glance_service_adminuri_proto: "{{ openstack_service_adminuri_proto |␣
↪default(glance_service_proto) }}"
glance_service_internaluri_proto: "{{ openstack_service_internaluri_proto |␣
↪default(glance_service_proto) }}"
glance_service_type: image
glance_service_description: "Glance Image Service"
glance_service_user_name: glance
glance_service_project_name: service
glance_service_project_domain_id: default
glance_service_user_domain_id: default
glance_service_publicuri: "{{ glance_service_publicuri_proto }}://{{ external_
↪lb_vip_address }}:{{ glance_service_port }}"
glance_service_publicurl: "{{ glance_service_publicuri }}"
glance_service_internaluri: "{{ glance_service_internaluri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ glance_service_port }}"
glance_service_internalurl: "{{ glance_service_internaluri }}"
glance_service_adminuri: "{{ glance_service_adminuri_proto }}://{{ internal_
↪lb_vip_address }}:{{ glance_service_port }}"
glance_service_adminurl: "{{ glance_service_adminuri }}"

## Enable automatic parsing of X-Forwarded-Proto, etc. headers passed by the
## load balancer.
glance_proxy_headers_parsing: True

## Swift Options
glance_swift_store_auth_address: "{{ keystone_service_internalurl }}"
glance_swift_store_auth_insecure: "{{ keystone_service_internaluri_insecure }}
↪"
```

(continues on next page)

(continued from previous page)

```
glance_swift_store_auth_version: 3
glance_swift_store_user_domain: default
glance_swift_store_project_domain: default
glance_swift_store_user: "service:{{ glance_service_user_name }}"
glance_swift_store_key: "{{ glance_service_password }}"
glance_swift_store_region: "{{ glance_service_region }}"
glance_swift_store_container: glance_images
glance_swift_store_endpoint_type: internalURL
# Set the swift_store_large_objects variables in MB
glance_swift_store_large_object_size: 5120
glance_swift_store_large_object_chunk_size: 200

## Keystone authentication middleware
glance_keystone_auth_plugin: password

## Glance config
glance_image_cache_max_size: 10737418240
glance_image_cache_stall_time: 86400

# CORS options
glance_cors_allowed_origin: "{{ (glance_show_multiple_locations | bool) |␣
↪ternary(openstack_service_publicuri_proto | default('http') + '://' +␣
↪external_lb_vip_address, None) }}"
# If ``glance_api_workers`` is unset the system will use half the number of␣
↪available VCPUS to
# compute the number of api workers to use.
# glance_api_workers: 16

## Cap the maximum number of threads / workers when a user value is␣
↪unspecified.
glance_api_threads_max: 16
glance_api_threads: "{{ [[(ansible_facts['processor_vcpus']//ansible_facts[
↪'processor_threads_per_core'])|default(1), 1] | max, glance_api_threads_
↪max] | min }}"

glance_task_executor: taskflow
glance_digest_algorithm: sha256
glance_http_keepalive: True

## Glance policy
glance_policy_file: policy.yaml
glance_policy_default_rule: default
glance_policy_dirs: policy.d


# Define nfs information to enable nfs shares as mounted directories for
# glance. The ``glance_nfs_client`` value is a list of dictionaries that must
# be filled out completely to enable the persistent NFS mounts.
glance_nfs_local_directory: "images"
```

(continues on next page)

(continued from previous page)

```
glance_nfs_client: []

# Example of the expected dict structure:
#
# glance_nfs_client:
#   - server: "127.0.0.1"                   ## Hostname or IP address of NFS␣
↪Server
#     remote_path: "/images"                ## Remote path from the NFS server
↪'s export
#     local_path: "/var/lib/glance/images"  ## Local path on machine
#     type: "nfs"                           ## This can be nfs or nfs4
#     options: "_netdev,auto"               ## Mount options
#     config_overrides: "{}"                ## Override dictionary for unit␣
↪file

## Policy vars
# Provide a list of access controls to update the default policy.json with.␣
↪These changes will be merged
# with the access controls in the default policy.json. E.g.
#glance_policy_overrides:
#  "add_image": ""
#  "delete_image": ""

## Ceph rbd Options
glance_ceph_client: glance
glance_rbd_store_pool: images
glance_rbd_store_user: '{{ glance_ceph_client }}'
glance_rbd_store_chunk_size: 8

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

# Common pip packages
glance_pip_packages:
  - "git+{{ glance_git_repo }}@{{ glance_git_install_branch }}#egg=glance"
  - cryptography
  - keystonemiddleware
  - os-brick
  - oslo.rootwrap
  - osprofiler
  - PyMySQL
  - pymemcache
  - python-cinderclient
  - python-glanceclient
  - python-keystoneclient
  - python-memcached
  - python-swiftclient
  - systemd-python
  - warlock
  - "{{ ('s3' in [ glance_default_store ] + glance_additional_stores) |␣
↪ternary('boto3', '') }}"
```

(continues on next page)

(continued from previous page)

```
# Specific pip packages provided by the user
glance_user_pip_packages: []

glance_optional_oslomsg_amqp1_pip_packages:
  - oslo.messaging[amqp1]

glance_api_init_overrides: {}

# With enabled uwsgi glance has broken functionality of
# the interoperable import feature (and maybe smth else)
# We don't use uwsgi if ceph is used to prevent chunking issues
glance_use_uwsgi: "{{ ('ceph' not in [ glance_default_store ] + glance_
↪additional_stores) }}"

## Service Names
glance_services:
  glance-api:
    group: glance_api
    service_name: glance-api
    init_config_overrides: "{{ glance_api_init_overrides }}"
    start_order: 1
    execstarts: "{{ glance_bin }}/glance-api"
    wsgi_app: "{{ glance_use_uwsgi }}"
    wsgi_name: glance-wsgi-api
    uwsgi_overrides: "{{ glance_api_uwsgi_ini_overrides }}"
    uwsgi_bind_address: "{{ glance_api_bind_address }}"
    uwsgi_port: "{{ glance_api_service_port }}"

# Glance uWSGI settings
glance_wsgi_processes_max: 16
glance_wsgi_processes: "{{ [[(ansible_facts['processor_vcpus']//ansible_facts[
↪'processor_threads_per_core'])|default(1), 1] | max * 2, glance_wsgi_
↪processes_max] | min }}"
glance_wsgi_threads: 1

## Tunable overrides
glance_glance_api_paste_ini_overrides: {}
glance_glance_api_conf_overrides: {}
glance_glance_cache_conf_overrides: {}
glance_glance_manage_conf_overrides: {}
glance_glance_scrubber_conf_overrides: {}
glance_glance_scheme_json_overrides: {}
glance_glance_swift_store_conf_overrides: {}
glance_policy_overrides: {}
glance_policy_content: {}
glance_api_uwsgi_ini_overrides: {}

# Specify path on the local filesystem for glance-image-import.conf
```

(continues on next page)

(continued from previous page)

```
#glance_glance_image_import_conf_location: /path/to/local/glance-image-import.
↪conf
```

CHAPTER THREE

DEPENDENCIES

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

CHAPTER

FOUR

EXAMPLE PLAYBOOK

```yaml
---
- name: Install glance server
  hosts: glance_all
  user: root
  roles:
    - { role: "os_glance", tags: [ "os-glance" ] }
  vars:
    external_lb_vip_address: 172.16.24.1
    internal_lb_vip_address: 192.168.0.1
    glance_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: `glance-install` and `glance-config`. The `glance-install` tag can be used to install and upgrade. The `glance-config` tag can be used to manage configuration.