OpenStack-Ansible Documentation: os_nova role

Release 18.1.0.dev428

OpenStack-Ansible Contributors

Mar 21, 2022

CONTENTS

CHAPTER

ONE

CONFIGURING THE COMPUTE (NOVA) SERVICE (OPTIONAL)

The Compute service (nova) handles the creation of virtual machines within an OpenStack environment. Many of the default options used by OpenStack-Ansible are found within `defaults/main.yml` within the nova role.

1.1 Availability zones

Deployers with multiple availability zones can set the `nova_nova_conf_overrides.DEFAULT.default_schedule_zone` Ansible variable to specify an availability zone for new requests. This is useful in environments with different types of hypervisors, where builds are sent to certain hardware types based on their resource requirements.

For example, if you have servers running on two racks without sharing the PDU. These two racks can be grouped into two availability zones. When one rack loses power, the other one still works. By spreading your containers onto the two racks (availability zones), you will improve your service availability.

1.2 Block device tuning for Ceph (RBD)

Enabling Ceph and defining `nova_libvirt_images_rbd_pool` changes two libvirt configurations by default:

- hw_disk_discard: `unmap`
- disk_cachemodes: `network=writeback`

Setting `hw_disk_discard` to `unmap` in libvirt enables discard (sometimes called TRIM) support for the underlying block device. This allows reclaiming of unused blocks on the underlying disks.

Setting `disk_cachemodes` to `network=writeback` allows data to be written into a cache on each change, but those changes are flushed to disk at a regular interval. This can increase write performance on Ceph block devices.

You have the option to customize these settings using two Ansible variables (defaults shown here):

```
nova_libvirt_hw_disk_discard: 'unmap'
nova_libvirt_disk_cachemodes: 'network=writeback'
```

You can disable discard by setting `nova_libvirt_hw_disk_discard` to `ignore`. The `nova_libvirt_disk_cachemodes` can be set to an empty string to disable `network=writeback`.

The following minimal example configuration sets nova to use the `ephemeral-vms` Ceph pool. The following example uses cephx authentication, and requires an existing `cinder` account for the `ephemeral-vms` pool:

```
nova_libvirt_images_rbd_pool: ephemeral-vms
ceph_mons:
  - 172.29.244.151
  - 172.29.244.152
  - 172.29.244.153
```

If you have a different Ceph username for the pool, use it as:

```
cinder_ceph_client: <ceph-username>
```

- The Ceph documentation for OpenStack has additional information about these settings.
- OpenStack-Ansible and Ceph Working Example

1.3 Config drive

By default, OpenStack-Ansible does not configure nova to force config drives to be provisioned with every instance that nova builds. The metadata service provides configuration information that is used by `cloud-init` inside the instance. Config drives are only necessary when an instance does not have `cloud-init` installed or does not have support for handling metadata.

A deployer can set an Ansible variable to force config drives to be deployed with every virtual machine:

```
nova_nova_conf_overrides:
  DEFAULT:
    force_config_drive: True
```

Certain formats of config drives can prevent instances from migrating properly between hypervisors. If you need forced config drives and the ability to migrate instances, set the config drive format to `vfat` using the `nova_nova_conf_overrides` variable:

```
nova_nova_conf_overrides:
  DEFAULT:
    config_drive_format: vfat
    force_config_drive: True
```

1.4 Libvirtd connectivity and authentication

By default, OpenStack-Ansible configures the libvirt daemon in the following way:

- TLS connections are enabled
- TCP plaintext connections are disabled
- Authentication over TCP connections uses SASL

You can customize these settings using the following Ansible variables:

```
# Enable libvirtd's TLS listener
nova_libvirtd_listen_tls: 1

# Disable libvirtd's plaintext TCP listener
nova_libvirtd_listen_tcp: 0

# Use SASL for authentication
nova_libvirtd_auth_tcp: sasl
```

1.5 Multipath

Nova supports multipath for iSCSI-based storage. Enable multipath support in nova through a configuration override:

```
nova_nova_conf_overrides:
  libvirt:
      iscsi_use_multipath: true
```

1.6 Shared storage and synchronized UID/GID

Specify a custom UID for the nova user and GID for the nova group to ensure they are identical on each host. This is helpful when using shared storage on Compute nodes because it allows instances to migrate without filesystem ownership failures.

By default, Ansible creates the nova user and group without specifying the UID or GID. To specify custom values for the UID or GID, set the following Ansible variables:

```
nova_system_user_uid = <specify a UID>
nova_system_group_gid = <specify a GID>
```

Warning: Setting this value after deploying an environment with OpenStack-Ansible can cause failures, errors, and general instability. These values should only be set once before deploying an OpenStack environment and then never changed.

tags openstack, nova, cloud, ansible

category *nix

This role will install the following Systemd services:

- nova-server
- nova-compute

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

CHAPTER

TWO

DEFAULT VARIABLES

```
# Enable/Disable barbican configurations
nova_barbican_enabled: False
# Enable/Disable designate configurations
nova_designate_enabled: False
# Notification topics for designate.
nova_notifications_designate: notifications_designate
# Enable/Disable ceilometer configurations
nova_ceilometer_enabled: False
# Enable/Disable nova versioned notification
nova_versioned_notification_enabled: False

nova_memcached_servers: "{{ memcached_servers }}"

## 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.
nova_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
nova_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((nova_service_setup_host == 'localhost
↪') | ternary(ansible_playbook_python, ansible_python['executable'])) }}"

# Set the host which will run compute initialization tasks such as checking
# for a compute node to be up and running cell discovery.
nova_conductor_setup_host: "{{ groups[nova_services['nova-conductor']['
↪'group']][0] }}"

# Set the package install state for distribution and pip packages
# Options are 'present' and 'latest'
nova_package_state: "latest"
nova_pip_package_state: "latest"

# Set installation method.
nova_install_method: "source"
nova_venv_python_executable: "{{ openstack_venv_python_executable |␣
↪default('python2') }}"

nova_git_repo: https://opendev.org/openstack/nova
nova_git_install_branch: master
```

(continues on next page)

(continued from previous page)

```
nova_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
nova_git_constraints:
  - "git+{{ nova_git_repo }}@{{ nova_git_install_branch }}#egg=nova"
  - "--constraint {{ nova_upper_constraints_url }}"
nova_pip_install_args: "{{ pip_install_options | default('') }}"

# Name of the virtual env to deploy into
nova_venv_tag: "{{ venv_tag | default('untagged') }}"
nova_bin: "{{ _nova_bin }}"

## Nova user information
nova_system_user_name: nova
nova_system_group_name: nova
nova_system_shell: /bin/bash
nova_system_comment: nova system user
nova_system_home_folder: "/var/lib/{{ nova_system_user_name }}"
nova_libvirt_save_path: "{{ nova_system_home_folder }}/save"

nova_lock_path: "/var/lock/nova"

nova_management_address: "127.0.0.1"

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

## Database info
nova_db_setup_host: "{{ openstack_db_setup_host | default('localhost') }}"
nova_db_setup_python_interpreter: "{{ openstack_db_setup_python_
↪interpreter | default((nova_db_setup_host == 'localhost') |␣
↪ternary(ansible_playbook_python, ansible_python['executable'])) }}"
nova_galera_address: "{{ galera_address | default('127.0.0.1') }}"
nova_galera_user: nova
nova_galera_database: nova
nova_db_max_overflow: 10
nova_db_max_pool_size: 120
nova_db_pool_timeout: 30
nova_galera_port: 3306
# Toggle whether nova connects via an encrypted connection
nova_galera_use_ssl: "{{ galera_use_ssl | default(False) }}"
# The path where to store the database server CA certificate
nova_galera_ssl_ca_cert: "{{ galera_ssl_ca_cert | default('/etc/ssl/certs/
↪galera-ca.pem') }}"

## DB API
```

(continues on next page)

(continued from previous page)

```
nova_api_galera_address: "{{ nova_galera_address }}"
nova_api_galera_user: nova_api
nova_api_galera_database: nova_api
nova_api_db_max_overflow: 10
nova_api_db_max_pool_size: 120
nova_api_db_pool_timeout: 30

## DB Cells
nova_cell0_database: "nova_cell0"
nova_cell1_name: "cell1"

## Oslo Messaging

# RPC
nova_oslomsg_rpc_host_group: "{{ oslomsg_rpc_host_group | default(
↪'rabbitmq_all') }}"
nova_oslomsg_rpc_setup_host: "{{ (nova_oslomsg_rpc_host_group in groups) |␣
↪ternary(groups[nova_oslomsg_rpc_host_group][0], 'localhost') }}"
nova_oslomsg_rpc_transport: "{{ oslomsg_rpc_transport | default('rabbit') }
↪}"
nova_oslomsg_rpc_servers: "{{ oslomsg_rpc_servers | default('127.0.0.1') }}
↪"
nova_oslomsg_rpc_port: "{{ oslomsg_rpc_port | default('5672') }}"
nova_oslomsg_rpc_use_ssl: "{{ oslomsg_rpc_use_ssl | default(False) }}"
nova_oslomsg_rpc_userid: nova
nova_oslomsg_rpc_vhost: /nova

# Notify
nova_oslomsg_notify_host_group: "{{ oslomsg_notify_host_group | default(
↪'rabbitmq_all') }}"
nova_oslomsg_notify_setup_host: "{{ (nova_oslomsg_notify_host_group in␣
↪groups) | ternary(groups[nova_oslomsg_notify_host_group][0], 'localhost
↪') }}"
nova_oslomsg_notify_transport: "{{ oslomsg_notify_transport | default(
↪'rabbit') }}"
nova_oslomsg_notify_servers: "{{ oslomsg_notify_servers | default('127.0.0.
↪1') }}"
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 }}"
nova_oslomsg_notify_password: "{{ nova_oslomsg_rpc_password }}"
nova_oslomsg_notify_vhost: "{{ nova_oslomsg_rpc_vhost }}"

## Qdrouterd info
# TODO(ansmith): Change structure when more backends will be supported
nova_oslomsg_amqp1_enabled: "{{ nova_oslomsg_rpc_transport == 'amqp' }}"

## Nova virtualization Types
# The nova_virt_types dictionary contains global overrides used for
#  specific compute types. Every variable inside of this dictionary
#  will become an ansible fact. This gives the user the option to set
#  or customize things based on their needs without having to redefine
#  this entire data structure. Every supported compute type will be
#  have its specific variable requirements set under its short name.
nova_virt_types:
```

(continues on next page)

(continued from previous page)

```
  ironic:
    nova_compute_driver: ironic.IronicDriver
    nova_reserved_host_memory_mb: 0
    nova_scheduler_tracks_instance_changes: False
  kvm:
    nova_compute_driver: libvirt.LibvirtDriver
    nova_reserved_host_memory_mb: 2048
    nova_scheduler_tracks_instance_changes: True
  qemu:
    nova_compute_driver: libvirt.LibvirtDriver
    nova_reserved_host_memory_mb: 2048
    nova_scheduler_tracks_instance_changes: True
    nova_cpu_mode: "none"


# If this is not set, then the playbook will try to guess it.
#nova_virt_type: kvm

# Enable Kernel Shared Memory (KSM)
nova_compute_ksm_enabled: False

#if set, nova_virt_type must be one of these:
nova_supported_virt_types:
  - qemu
  - kvm
  - ironic

## Nova Auth
nova_service_region: RegionOne
nova_service_project_name: "service"
nova_service_project_domain_id: default
nova_service_user_domain_id: default
nova_service_user_name: "nova"
nova_service_role_name: "admin"

## Keystone authentication middleware
nova_keystone_auth_plugin: password

## Nova enabled apis
nova_enabled_apis: "osapi_compute,metadata"

## Nova v2.1
nova_service_name: nova
nova_service_type: compute
nova_service_proto: http
nova_service_publicuri_proto: "{{ openstack_service_publicuri_proto |␣
↪default(nova_service_proto) }}"
nova_service_adminuri_proto: "{{ openstack_service_adminuri_proto |␣
↪default(nova_service_proto) }}"
nova_service_internaluri_proto: "{{ openstack_service_internaluri_proto |␣
↪default(nova_service_proto) }}"
nova_service_bind_address: "{{ openstack_service_bind_address | default('0.
↪0.0.0') }}"
nova_service_port: 8774
nova_service_description: "Nova Compute Service"
nova_service_publicuri: "{{ nova_service_publicuri_proto }}://{{ external_
↪lb_vip_address }}:{{ nova_service_port }}"
```

(continues on next page)

(continued from previous page)

```
nova_service_publicurl: "{{ nova_service_publicuri }}/v2.1"
nova_service_adminuri: "{{ nova_service_adminuri_proto }}://{{ internal_lb_
↪vip_address }}:{{ nova_service_port }}"
nova_service_adminurl: "{{ nova_service_adminuri }}/v2.1"
nova_service_internaluri: "{{ nova_service_internaluri_proto }}://{{␣
↪internal_lb_vip_address }}:{{ nova_service_port }}"
nova_service_internalurl: "{{ nova_service_internaluri }}/v2.1"

## Nova spice
nova_spice_html5proxy_base_proto: "{{ openstack_service_publicuri_proto |␣
↪default('http') }}"
nova_spice_html5proxy_base_port: 6082
nova_spice_html5proxy_base_uri: "{{ nova_spice_html5proxy_base_proto }}://{
↪{ external_lb_vip_address }}:{{ nova_spice_html5proxy_base_port }}"
nova_spice_html5proxy_base_url: "{{ nova_spice_html5proxy_base_uri }}/
↪spice_auto.html"
nova_spice_console_agent_enabled: "{{ ansible_architecture != 'aarch64' }}"
nova_spicehtml5_git_repo: "{{ spicehtml5_git_repo | default('https://
↪gitlab.freedesktop.org/spice/spice-html5.git') }}"
nova_spicehtml5_git_install_branch: "{{ spicehtml5_git_install_branch |␣
↪default('master') }}"

## Nova novnc
nova_novncproxy_proto: "{{ openstack_service_publicuri_proto | default(
↪'http') }}"
nova_novncproxy_port: 6080
nova_novncproxy_host: "{{ openstack_service_bind_address | default('0.0.0.0
↪') }}"
nova_novncproxy_base_uri: "{{ nova_novncproxy_proto }}://{{ external_lb_
↪vip_address }}:{{ nova_novncproxy_port }}"
nova_novncproxy_base_url: "{{ nova_novncproxy_base_uri }}/vnc_lite.html"
nova_novncproxy_vncserver_proxyclient_address: "{{ (nova_management_
↪address == 'localhost') | ternary('127.0.0.1', nova_management_address) }
↪}"
nova_novncproxy_vncserver_listen: "{{ (nova_management_address ==
↪'localhost') | ternary('127.0.0.1', nova_management_address) }}"
nova_novncproxy_agent_enabled: True
nova_novncproxy_git_repo: "{{ novncproxy_git_repo | default('https://
↪github.com/novnc/noVNC') }}"
nova_novncproxy_git_install_branch: "{{ novncproxy_git_install_branch |␣
↪default('master') }}"

## Nova serialconsole
nova_serialconsoleproxy_proto: "ws"
nova_serialconsoleproxy_port: 6083
nova_serialconsoleproxy_port_range: 10000:20000
nova_serialconsoleproxy_base_uri: "{{ nova_serialconsoleproxy_proto }}://{
↪{ external_lb_vip_address }}:{{ nova_serialconsoleproxy_port }}"
nova_serialconsoleproxy_base_url: "{{ nova_serialconsoleproxy_base_uri }}"
nova_serialconsoleproxy_serialconsole_proxyserver_proxyclient_address: "{{␣
↪ansible_host }}"
nova_serialconsoleproxy_enabled: True

## Nova metadata
nova_metadata_proxy_enabled: "{{ nova_network_services[nova_network_type][
↪'metadata_proxy_enabled'] | bool }}"
```

(continues on next page)

(continued from previous page)

```
nova_metadata_bind_address: "{{ openstack_service_bind_address | default(
↪'0.0.0.0') }}"
nova_metadata_port: 8775

## Nova compute
nova_nested_virt_enabled: False

# Uwsgi settings
nova_wsgi_processes_max: 16
nova_wsgi_processes: "{{ [[ansible_processor_vcpus|default(1), 1] | max *␣
↪2, nova_wsgi_processes_max] | min }}"
nova_wsgi_threads: 1

## Nova libvirt
# Warning: If nova_libvirt_inject_key or nova_libvirt_inject_password are␣
↪enabled for Ubuntu compute hosts
# then the kernel will be made readable to nova user (a requirement for␣
↪libguestfs).
# See Launchpad bugs 1507915 (Nova), 759725 (Ubuntu), and http://
↪libguestfs.org/guestfs-faq.1.html
nova_libvirt_inject_key: False
# inject partition options:
#  -2 => disable, -1 => inspect (libguestfs only), 0 => not partitioned, >
↪0 => partition number
nova_libvirt_inject_partition: -2
nova_libvirt_inject_password: False
nova_libvirt_disk_cachemodes: '{{ (nova_libvirt_images_rbd_pool | length >␣
↪0) | ternary("network=writeback", "") }}'
nova_libvirt_hw_disk_discard: '{{ (nova_libvirt_images_rbd_pool | length >␣
↪0) | ternary("unmap", "ignore") }}'
nova_libvirt_live_migration_inbound_addr: '{{ (nova_management_address ==
↪"localhost") | ternary("127.0.0.1", nova_management_address) }}'

## Nova console
nova_console_agent_enabled: True
# Set the console type. Presently the only options are ["spice", "novnc",
↪"serialconsole"].
nova_console_type: "{{ (ansible_architecture == 'aarch64') | ternary(
↪'serialconsole', 'novnc') }}"

# Nova console ssl info, presently only used by novnc console type
nova_console_ssl_dir: "/etc/nova/ssl"
nova_console_ssl_cert: "{{ nova_console_ssl_dir }}/nova-console.pem"
nova_console_ssl_key: "{{ nova_console_ssl_dir }}/nova-console.key"

# Set to true when terminating SSL/TLS at a load balancer
nova_external_ssl: false

# External SSL forwarding proto
nova_secure_proxy_ssl_header: HTTP_X_FORWARDED_PROTO

## Nova global config
nova_image_cache_manager_interval: 0

# Comma separated list of Glance API servers
nova_glance_api_servers: "{{ (glance_service_internalurl | default('http://
↪localhost')) | urlsplit('scheme') ~ '://' ~ (glance_service_internalurl
↪| default('http://localhost')) | urlsplit('netloc') }}"
```

(continues on next page)

(continued from previous page)

```
nova_network_type: linuxbridge

nova_network_services:
  linuxbridge:
    use_forwarded_for: False
    metadata_proxy_enabled: True
  nuage:
    use_forwarded_for: True
    metadata_proxy_enabled: True
    ovs_bridge: alubr0
  calico:
    use_forwarded_for: True
    metadata_proxy_enabled: False
  nsx:
    use_forwarded_for: True
    metadata_proxy_enabled: True
    ovs_bridge: nsx-managed

# Nova Scheduler
nova_cpu_allocation_ratio: 2.0
nova_disk_allocation_ratio: 1.0
nova_max_io_ops_per_host: 10
nova_ram_allocation_ratio: 1.0
nova_ram_weight_multiplier: 5.0
nova_reserved_host_disk_mb: 2048

nova_scheduler_host_subset_size: 10
nova_scheduler_max_attempts: 5
nova_scheduler_default_filters:
  - AvailabilityZoneFilter
  - ComputeFilter
  - AggregateNumInstancesFilter
  - AggregateIoOpsFilter
  - ComputeCapabilitiesFilter
  - ImagePropertiesFilter
  - ServerGroupAntiAffinityFilter
  - ServerGroupAffinityFilter
  - NUMATopologyFilter

nova_scheduler_extra_filters: []

# This should be tuned depending on the number of compute hosts present in␣
↪the
# deployment. Large clouds may wish to disable automatic discovery by␣
↪setting
# this value to -1.
nova_discover_hosts_in_cells_interval: "{{ 300 if groups['nova_compute'] |␣
↪length > 10 else 60 }}"

# If you want to regenerate the nova users SSH keys, on each run, set this␣
↪var to True
# Otherwise keys will be generated on the first run and not regenerated␣
↪each run.
nova_recreate_keys: False
```

(continues on next page)

(continued from previous page)

```
# Define nfs information to enable nfs shares as mounted directories for
# nova. The ``nova_nfs_client`` value is a list of dictionaries that must
# be filled out completely to enable the persistent NFS mounts.
#
# Example of the expected dict structure:
# nova_nfs_client:
#   - server: "127.0.0.1"                    ## Hostname or IP address of␣
↪NFS Server
#     remote_path: "/instances"              ## Remote path from the NFS␣
↪server's export
#     local_path: "/var/lib/nova/instances" ## Local path on machine
#     type: "nfs"                            ## This can be nfs or nfs4
#     options: "_netdev,auto"                ## Mount options
#     config_overrides: "{}"                 ## Override dictionary for␣
↪unit file
nova_nfs_client: []

# Nova Ceph rbd
# Enble and define nova_libvirt_images_rbd_pool to use rbd as nova backend
#nova_libvirt_images_rbd_pool: vms
nova_libvirt_images_rbd_pool: ''
nova_ceph_client: '{{ cinder_ceph_client }}'
# TODO(odyssey4me) - the uuid should be removed, there should be no␣
↪defaults for secrets
nova_ceph_client_uuid: 517a4663-3927-44bc-9ea7-4a90e1cd4c66

# Enabled upstream if RBD is in use on cinder backends, which requires that
# ceph be deployed on the nova compute hosts to enable volume backed␣
↪instances.
nova_cinder_rbd_inuse: False

# Used to determine if we need a Ceph client
nova_rbd_inuse: "{{ (nova_libvirt_images_rbd_pool | length > 0) or (nova_
↪cinder_rbd_inuse | bool) }}"

## General Nova configuration
# If ``nova_conductor_workers`` is unset the system will use half the␣
↪number of available VCPUS to
# compute the number of api workers to use.
# nova_conductor_workers: 16

# If ``nova_scheduler_workers`` is unset the system will use half the␣
↪number of available VCPUS to
# compute the number of api workers to use.
# nova_scheduler_workers: 16

## Cap the maximun number of threads / workers when a user value is␣
↪unspecified.
nova_api_threads_max: 16
nova_api_threads: "{{ [[(ansible_processor_vcpus//ansible_processor_
↪threads_per_core)|default(1), 1] | max * 2, nova_api_threads_max] | min }
↪}"

## Policy vars
# Provide a list of access controls to update the default policy.json with.
↪ These changes will be merged
```

(continues on next page)

(continued from previous page)

```
# with the access controls in the default policy.json. E.g.
#nova_policy_overrides:
#  "compute:create": ""
#  "compute:create:attach_network": ""

nova_service_in_ldap: false

## libvirtd config options
nova_libvirtd_listen_tls: 0
nova_libvirtd_listen_tcp: 0
nova_libvirtd_auth_tcp: sasl
nova_libvirtd_debug_log_filters: "3:remote 4:event 3:json 3:rpc"

nova_api_metadata_init_overrides: {}
nova_api_os_compute_init_overrides: {}
nova_compute_init_overrides: {}
nova_conductor_init_overrides: {}
nova_novncproxy_init_overrides: {}
nova_scheduler_init_overrides: {}
nova_spicehtml5proxy_init_overrides: {}
nova_serialproxy_init_overrides: {}

## Service Name-Group Mapping
nova_services:
  nova-api-metadata:
    group: nova_api_metadata
    service_name: nova-api-metadata
    init_config_overrides: "{{ nova_api_metadata_init_overrides }}"
    start_order: 5
    wsgi_app: True
    uwsgi_overrides: "{{ nova_api_metadata_uwsgi_ini_overrides }}"
    uwsgi_bind_address: "{{ nova_metadata_bind_address }}"
    uwsgi_port: "{{ nova_metadata_port }}"
    wsgi_name: nova-metadata-wsgi
  nova-api-os-compute:
    group: nova_api_os_compute
    service_name: nova-api-os-compute
    init_config_overrides: "{{ {'Install': {'Alias': 'nova-api.service'}}␣
↪| combine(nova_api_os_compute_init_overrides) }}"
    start_order: 4
    wsgi_app: True
    uwsgi_overrides: "{{ nova_api_os_compute_uwsgi_ini_overrides }}"
    uwsgi_bind_address: "{{ nova_service_bind_address }}"
    uwsgi_port: "{{ nova_service_port }}"
    wsgi_name: nova-api-wsgi
  nova-compute:
    group: nova_compute
    service_name: nova-compute
    init_config_overrides: "{{ nova_compute_init_overrides }}"
    start_order: 6
    execstarts: "{{ nova_bin }}/nova-compute"
    execreloads: "/bin/kill -HUP $MAINPID"
    after_targets:
      - libvirtd.service
      - syslog.target
      - network.target
```

(continues on next page)

(continued from previous page)

```
  nova-conductor:
    group: nova_conductor
    service_name: nova-conductor
    init_config_overrides: "{{ nova_conductor_init_overrides }}"
    start_order: 2
    execstarts: "{{ nova_bin }}/nova-conductor"
    execreloads: "/bin/kill -HUP $MAINPID"
  nova-novncproxy:
    group: nova_console
    service_name: nova-novncproxy
    init_config_overrides: "{{ nova_novncproxy_init_overrides }}"
    condition: "{{ nova_console_type == 'novnc' }}"
    start_order: 5
    execstarts: "{{ nova_bin }}/nova-novncproxy"
  nova-scheduler:
    group: nova_scheduler
    service_name: nova-scheduler
    init_config_overrides: "{{ nova_scheduler_init_overrides }}"
    start_order: 3
    execstarts: "{{ nova_bin }}/nova-scheduler"
    execreloads: "/bin/kill -HUP $MAINPID"
  nova-spicehtml5proxy:
    group: nova_console
    service_name: nova-spicehtml5proxy
    init_config_overrides: "{{ {'Install': {'Alias': 'nova-spiceproxy.
↪service'}} | combine(nova_spicehtml5proxy_init_overrides) }}"
    condition: "{{ nova_console_type == 'spice' }}"
    start_order: 5
    execstarts: "{{ nova_bin }}/nova-spicehtml5proxy"
  nova-serialconsole-proxy:
    group: nova_console
    service_name: nova-serialproxy
    init_config_overrides: "{{ nova_serialproxy_init_overrides }}"
    condition: "{{ nova_console_type == 'serialconsole' }}"
    start_order: 5
    execstarts: "{{ nova_bin }}/nova-serialproxy"

nova_novnc_pip_packages:
  - websockify

nova_compute_ironic_pip_packages:
  - python-ironicclient

# Common pip packages
nova_pip_packages:
  - nova
  - osprofiler
  - PyMySQL
  - pymemcache
  - python-memcached
  - systemd-python

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

nova_optional_oslomsg_amqp1_pip_packages:
```

(continues on next page)

(continued from previous page)

```
  - oslo.messaging[amqp1]

nova_qemu_user: libvirt-qemu
nova_qemu_group: kvm

# Define the following variable to drop a replacement
# file for /etc/libvirt/qemu.conf
#qemu_conf_dict: {}

## Tunable overrides
nova_nova_conf_overrides: {}
nova_rootwrap_conf_overrides: {}
nova_api_paste_ini_overrides: {}
nova_policy_overrides: {}
nova_vendor_data_overrides: {}
nova_api_metadata_uwsgi_ini_overrides: {}
nova_api_os_compute_uwsgi_ini_overrides: {}

# Enabled vGPU Types - dict defining 'type' and 'address' (optional) of␣
↪vGPU
# an address is only required when supporting more than one physical GPU␣
↪on the host
# Example 1:
# nova_enabled_vgpu_types:
#   - type: nvidia-35
#
# Example 2:
# nova_enabled_vgpu_types:
#   - type: nvidia-35
#     address: <device address>
#   - type: nvidia-36
#     address: <another device address>
nova_enabled_vgpu_types: {}

# PCI devices passthrough to nova
# For SR-IOV please use:
#   nova_pci_passthrough_whitelist: '{ "physical_network": "<ml2 network␣
↪name>", "devname": "<physical nic>" }'
#   Example:
#     nova_pci_passthrough_whitelist: '{ "physical_network": "physnet1",
↪"devname": "p1p1" }'
nova_pci_passthrough_whitelist: {}
# PCI alias,
# Example:
# nova_pci_alias:
#  - '{ "name": "card-alias1", "product_id": "XXXX", "vendor_id": "XXXX" }'
#  - '{ "name": "card-alias2", "product_id": "XXXY", "vendor_id": "XXXY" }'
nova_pci_alias: []
```

CHAPTER

THREE

DEPENDENCIES

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

CHAPTER

FOUR

EXAMPLE PLAYBOOK

```
---
- name: Installation and setup of Nova
  hosts: nova_all
  user: root
  roles:
    - { role: "os_nova", tags: [ "os-nova" ] }
  vars:
    nova_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: `nova-install` and `nova-config`

The `nova-install` tag can be used to install and upgrade.

The `nova-config` tag can be used to manage configuration.

CHAPTER

SEVEN

CPU PLATFORM COMPATIBILITY

This role supports multiple CPU architecture types. At least one repo_build node must exist for each CPU type that is in use in the deployment.

Currently supported CPU architectures:

- x86_64 / amd64
- ppc64le

At this time, ppc64le is only supported for the Compute node type. It can not be used to manage the OpenStack-Ansible management nodes.

CHAPTER

EIGHT

COMPUTE DRIVER COMPATIBILITY

This role supports multiple nova compute driver types. The following compute drivers are supported:

- libvirt (default)
- ironic

The driver type is automatically detected by the OpenStack Ansible Nova role for the following compute driver types:

- libvirt (kvm / qemu)

Any mix and match of compute node types can be used for those platforms, except for ironic.

The `nova_virt_type` may be set in `/etc/openstack_deploy/user_variables.yml`, for example:

```
nova_virt_type: ironic
```

You can set `nova_virt_type` per host by using `host_vars` in `/etc/openstack_deploy/openstack_user_config.yml`. For example:

```
compute_hosts:
 aio1:
   ip: 172.29.236.100
   host_vars:
     nova_virt_type: ironic
```

If `nova_virt_type` is set in `/etc/openstack_deploy/user_variables.yml`, all nodes in the deployment are set to that hypervisor type. Setting `nova_virt_type` in both `/etc/openstack_deploy/user_variables.yml` and `/etc/openstack_deploy/openstack_user_config.yml` will always result in the value specified in `/etc/openstack_deploy/user_variables.yml` being set on all hosts.