OpenStack-Ansible Documentation: os_horizon role

Release 18.1.0.dev304

OpenStack-Ansible Contributors

May 12, 2024

CONTENTS

CONFIGURING THE DASHBOARD (HORIZON) (OPTIONAL)

Customize your horizon deployment in `/etc/openstack_deploy/user_variables.yml`.

1.1 Securing horizon communication with SSL certificates

The OpenStack-Ansible project provides the ability to secure Dashboard (horizon) communications with self-signed or user-provided SSL certificates.

Refer to Securing services with SSL certificates for available configuration options.

1.2 Configuring a horizon customization module

OpenStack-Ansible supports deployment of a horizon customization module. After building your customization module, configure the `horizon_customization_module` variable with a path to your module.

```
horizon_customization_module: /path/to/customization_module.py
```

This Ansible role installs and configures OpenStack Horizon served by the Apache webserver. Horizon is configured to use Galera for session caching and Memcached for other caching.

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

DEFAULT VARIABLES

```
## 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.
horizon_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
horizon_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((horizon_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'
horizon_package_state: "{{ package_state | default('latest') }}"

# Set installation method.
horizon_install_method: "{{ service_install_method | default('source') }}"
horizon_venv_python_executable: "{{ openstack_venv_python_executable |
↪default('python3') }}"

horizon_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"

horizon_venv_extra_constraints: []

## The git source/branch for Horizon
horizon_git_repo: https://opendev.org/openstack/horizon
horizon_git_track_branch: master
horizon_git_install_branch: "{{ horizon_git_track_branch }}"

# Enable or disable apache. If disabled a basic uwsgi deployment will be
↪setup for this service.
horizon_use_uwsgi: false

## The git source/branch for the Adjutant UI plugin
```

```yaml
adjutant_dashboard_git_repo: https://opendev.org/openstack/adjutant-ui
adjutant_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Barbican UI plugin
barbican_dashboard_git_repo: https://opendev.org/openstack/barbican-ui
barbican_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Blazar UI plugin
blazar_dashboard_git_repo: https://opendev.org/openstack/blazar-dashboard
blazar_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for Cloudkitty UI plugin
cloudkitty_dashboard_git_repo: https://opendev.org/openstack/cloudkitty-
↪dashboard
cloudkitty_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Magnum UI plugin
magnum_dashboard_git_repo: https://opendev.org/openstack/magnum-ui
magnum_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Manila UI plugin
manila_dashboard_git_repo: https://opendev.org/openstack/manila-ui.git
manila_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Masakari UI plugin
masakari_dashboard_git_repo: https://opendev.org/openstack/masakari-dashboard
masakari_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Mistral UI plugin
mistral_dashboard_git_repo: https://opendev.org/openstack/mistral-dashboard.
↪git
mistral_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Murano UI plugin
murano_dashboard_git_repo: https://opendev.org/openstack/murano-dashboard
murano_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Octavia UI plugin
octavia_dashboard_git_repo: https://opendev.org/openstack/octavia-dashboard
octavia_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Designate UI plugin
designate_dashboard_git_repo: https://opendev.org/openstack/designate-
↪dashboard
designate_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Tacker UI plugin
tacker_dashboard_git_repo: https://opendev.org/openstack/tacker-horizon
tacker_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"
```

```yaml
## The git source/branch for the Trove UI plugin
trove_dashboard_git_repo: https://opendev.org/openstack/trove-dashboard
trove_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Heat UI plugin
heat_dashboard_git_repo: https://opendev.org/openstack/heat-dashboard
heat_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Ironic UI plugin
ironic_dashboard_git_repo: https://opendev.org/openstack/ironic-ui
ironic_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Watcher UI plugin
watcher_dashboard_git_repo: https://opendev.org/openstack/watcher-dashboard
watcher_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The git source/branch for the Zun UI plugin
zun_dashboard_git_repo: https://opendev.org/openstack/zun-ui
zun_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

# The git source/branch for the Neutron VPNaaS UI plugin
neutron_vpnaas_dashboard_git_repo: https://opendev.org/openstack/neutron-
↪vpnaas-dashboard
neutron_vpnaas_dashboard_git_install_branch: "{{ horizon_git_track_branch }}"

## The packages to build from source
horizon_git_constraints:
  - --constraint {{ horizon_upper_constraints_url }}

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

# Name of the virtual env to deploy into
horizon_venv_tag: "{{ venv_tag | default('untagged') }}"
horizon_bin: "{{ _horizon_bin }}"

## System info
horizon_system_user_name: horizon
horizon_system_group_name: horizon

horizon_system_shell: /bin/false
horizon_system_comment: horizon system user
horizon_system_user_home: "/var/lib/{{ horizon_system_user_name }}"

## Service Type and Data
horizon_service_region: "{{ service_region | default('RegionOne') }}"
horizon_service_name: horizon
## Session configuration
# Specifies the timespan in seconds inactivity, until a user is considered as
```

```
# logged out
horizon_session_engine: 'django.contrib.sessions.backends.cache'
horizon_session_caches:
  default:
    BACKEND: 'django.core.cache.backends.memcached.MemcachedCache'
    LOCATION: "{{ horizon_memcached_servers.split(',') }}"
horizon_session_timeout: 1800

## Horizon Help URL Path
horizon_help_url: https://docs.openstack.org/horizon/latest/user/

## Horizon ALLOWED_HOSTS
horizon_allowed_hosts:
  - '*'

## Installation directories
# When horizon_lib_dir is not defined, it will be detected automatically
#horizon_lib_dir: ""
horizon_lib_wsgi_file: "{{ horizon_lib_dir }}/openstack_dashboard/wsgi.py"

horizon_endpoint_type: internalURL

horizon_server_name: "{{ ansible_facts['fqdn'] | default('horizon') }}"

horizon_apache_servertokens: "Prod"
horizon_apache_serversignature: "Off"
horizon_log_level: info
# It's combined log format without datetime, since it's already present in␣
↪journald
horizon_apache_custom_log_format: '"%h %l %u \"%r\" %>s %b \"%{Referer}i\" \"%
↪{User-agent}i\""'
horizon_dropdown_max_items: 30
horizon_instance_log_length: 35
horizon_overview_days_range: 1

# Valid options are "direct", "legacy" and "off".
# Direct mode uploads images directly to glance, but requires CORS
# to be configured for glance. Legacy will use horizon container as
# a proxy for the image before uploading it to the glance, but do not
# require any extra configuration
horizon_images_upload_mode: direct

horizon_images_allow_location: False
horizon_time_zone: UTC
horizon_enforce_password_check: False
horizon_disable_password_reveal: False
horizon_enable_password_retrieve: False
horizon_enable_password_autocomplete: False
horizon_enable_heatstack_user_pass: True
```

```
# If nova_libvirt_inject_password is set to True, then this can also be␣
↪enabled:
horizon_can_set_password: False
horizon_enable_cinder_backup: False
# Enables IPv6 support in Horizon, such as managing network subnets
horizon_enable_ipv6: True
# Enables router support in Horizon, disable if you don't have Neutron L3 agent
horizon_enable_router: True

# Disable/Enable simplified floating IP address management for deployments␣
↪with
# multiple floating IP pools or complex network requirements.
horizon_simple_ip_management: True

# To enable ha router support in horizon set to True
horizon_enable_ha_router: False

# Provide default DNS servers to use when a subnet is created.
horizon_default_dns_nameservers: []

# Provide list of network types that are available for creation
horizon_network_provider_types: "{{ neutron_ml2_drivers_type | default(
↪'geneve,vlan,flat') }}"

# DISALLOW_IFRAME_EMBED can be used to prevent Horizon from being embedded
# within an iframe. Legacy browsers are still vulnerable to a Cross-Frame
# Scripting (XFS) vulnerability, so this option allows extra security␣
↪hardening
# where iframes are not used in deployment. Default setting is True.
# For more information see:
# http://tinyurl.com/anticlickjack
horizon_disallow_iframe_embed: True

# WSGI tuning parameters
# horizon_wsgi_processes: 4
# horizon_wsgi_threads: 4

## Cap the maximun number of threads / workers when a user value is␣
↪unspecified.
horizon_wsgi_threads_max: 16
horizon_wsgi_threads: "{{ [[ansible_facts['processor_vcpus']|default(2) // 2,␣
↪1] | max, horizon_wsgi_threads_max] | min }}"

## Horizon SSL
horizon_ssl_cert: /etc/ssl/certs/horizon.pem
horizon_ssl_key: /etc/ssl/private/horizon.key
horizon_ssl_ca_cert: /etc/ssl/certs/horizon-ca.pem
horizon_ssl_protocol: "{{ ssl_protocol | default('ALL -SSLv2 -SSLv3 -TLSv1 -
↪TLSv1.1') }}"
```

```
# TLS v1.2 and below
horizon_ssl_cipher_suite_tls12: "{{ horizon_ssl_cipher_suite | default(ssl_
↪cipher_suite | default('ECDH+AESGCM:ECDH+CHACHA20:ECDH+AES256:ECDH+AES128:!
↪aNULL:!SHA1:!AESCCM')) }}"
# TLS v1.3
horizon_ssl_cipher_suite_tls13: "{{ ssl_cipher_suite_tls13 | default('TLS_AES_
↪128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256') }}"
# if using a self-signed certificate, set this to true to regenerate it
horizon_ssl_self_signed_regen: false
horizon_ssl_self_signed_subject: "/C=US/ST=Texas/L=San Antonio/O=IT/CN={{
↪horizon_server_name }}/subjectAltName=IP.1={{ external_lb_vip_address }}"

# Set these variables to deploy custom certificates
# horizon_user_ssl_cert: <path to cert on ansible deployment host>
# horizon_user_ssl_key: <path to cert on ansible deployment host>
# horizon_user_ssl_ca_cert: <path to cert on ansible deployment host>

# Toggle whether horizon should be served via SSL
horizon_enable_ssl: "{{ (haproxy_ssl | default(True)) | bool }}"

# Toggle whether horizon is served via an external device, like a load
# balancer. This enables the use of the horizon_secure_proxy_ssl_header
# in the web server configuration.
# Note (odyssey4me):
# This variable is actually badly named, as it applies
# settings which have nothing to do with SSL.
horizon_external_ssl: "{{ (openstack_external_ssl | default(False)) | bool }}"

# Set this to the header that your device sets when doing ssl termination
# Note (odyssey4me):
# This variable is actually badly named, as it applies
# settings which have nothing to do with SSL.
horizon_secure_proxy_ssl_header: "X-Forwarded-Proto"
horizon_secure_proxy_ssl_header_django: "HTTP_{{ horizon_secure_proxy_ssl_
↪header | replace('-', '_') | upper }}"

# For multiple regions uncomment this configuration, and
# add the extra endpoints below the first list item.
# horizon_available_regions:
#   - { url: "{{ keystone_service_internalurl }}", name: "{{ keystone_service_
↪region }}" }
#   - { url: "http://cluster1.example.com:5000/v2.0", name: "RegionTwo" }

## Horizon's keystone endpoint settings
horizon_keystone_endpoint: "{{ keystone_service_internalurl }}"

## Horizon Multi-Domain Support
# these variables should only be changed if horizon_keystone_endpoint is a
↪Keystone v3 API endpoint
```

```
horizon_keystone_multidomain_support: False
# It is strongly advised NOT to enable dropdown for public clouds, as␣
↪advertising enabled domains
# to unauthenticated customers irresponsibly exposes private information.
horizon_keystone_multidomain_dropdown: False


horizon_keystone_default_domain: Default


# Option to set the available domains to choose from. This is
# a list of pairs whose first value is the domain name and the
# second is the display name.
horizon_keystone_multidomain_choices: "(('{{ horizon_keystone_default_domain }
↪}', '{{ horizon_keystone_default_domain }}'),)"


# Enable/disable v2 openrc file download in horizon.
horizon_show_keystone_v2_rc: True


### Set the cacert pem for Keystone if you'd like Horizon to verify it.
# horizon_cacert_pem: /path/to/cacert.pem


## alternatively, you can set horizon to turn off ssl verification for␣
↪Keystone
horizon_ssl_no_verify: "{{ (keystone_service_adminuri_insecure | bool or␣
↪keystone_service_internaluri_insecure | bool) | default(false) }}"


## Horizon Cross-Site Request Forgery Trusted Hosts
# add a list of domains that are trusted when evaluated requests for Cross-
↪Site Request Forgery
# This is useful when terminating SSL outside of the cloud on a domain that␣
↪isn't directly tied
# to the hosts that are operating the cloud.
horizon_ssl_csrf_trusted_origins: []


## The role which Horizon should use as a default for users
horizon_default_role_name: _member_


## Launch instance
horizon_launch_instance_legacy: False
horizon_launch_instance_ng: True
horizon_launch_instance_defaults:
  config_drive: False
  enable_scheduler_hints: True
  disable_image: False
  disable_instance_snapshot: False
  disable_volume: False
  disable_volume_snapshot: False
  create_volume: True
  hide_create_volume: False
```

```yaml
## Adjutant UI Panel
horizon_enable_adjutant_ui: "{{ (groups['adjutant_all'] is defined) and
→(groups['adjutant_all'] | length > 0) }}"

## Barbican UI Panel
### Barbican UI is just a cookie-cutter scaffolding and does not provide any
→functionality at this time.
horizon_enable_barbican_ui: False

## Blazar UI Panel
horizon_enable_blazar_ui: "{{ (groups['blazar_all'] is defined) and (groups[
→'blazar_all'] | length > 0) }}"

## Cloudkitty UI Panel
horizon_enable_cloudkitty_ui: "{{ (groups['cloudkitty_all'] is defined) and
→(groups['cloudkitty_all'] | length > 0) }}"

## Designate UI Panel
horizon_enable_designate_ui: "{{ (groups['designate_all'] is defined) and
→(groups['designate_all'] | length > 0) }}"

## Heat UI Panel
horizon_enable_heat_ui: "{{ (groups['heat_all'] is defined) and (groups['heat_
→all'] | length > 0) }}"

## Ironic UI Panel
horizon_enable_ironic_ui: "{{ (groups['ironic_all'] is defined) and (groups[
→'ironic_all'] | length > 0) }}"

## Magnum UI Panel
horizon_enable_magnum_ui: "{{ (groups['magnum_all'] is defined) and (groups[
→'magnum_all'] | length > 0) }}"

## Masakari UI Panel
horizon_enable_masakari_ui: "{{ (groups['masakari_all'] is defined) and
→(groups['masakari_all'] | length > 0) }}"

## Manila UI Panel
horizon_enable_manila_ui: "{{ (groups['manila_all'] is defined) and (groups[
→'manila_all'] | length > 0) }}"

## Mistral Ui Panel
horizon_enable_mistral_ui: "{{ (groups['mistral_all'] is defined) and (groups[
→'mistral_all'] | length > 0) }}"

## Murano UI Panel
horizon_enable_murano_ui: "{{ (groups['murano_all'] is defined) and (groups[
→'murano_all'] | length > 0) }}"
```

```yaml
## Neutron features to enable
horizon_enable_neutron_vpnaas: "{{ neutron_plugin_base is defined and 'vpnaas
↪' in neutron_plugin_base }}"

## Octavia UI Panel
horizon_enable_octavia_ui: "{{ (groups['octavia_all'] is defined) and (groups[
↪'octavia_all'] | length > 0) }}"

## Sahara UI Panel
horizon_enable_sahara_ui: "{{ (groups['sahara_all'] is defined) and (groups[
↪'sahara_all'] | length > 0) }}"

## Tacker UI Panel
horizon_enable_tacker_ui: "{{ (groups['tacker_all'] is defined) and (groups[
↪'tacker_all'] | length > 0) }}"

## Trove UI Panel
horizon_enable_trove_ui: "{{ (groups['trove_all'] is defined) and (groups[
↪'trove_all'] | length > 0) }}"

## Watcher UI Panel
horizon_enable_watcher_ui: False

## Zun UI Panel
horizon_enable_zun_ui: "{{ (groups['zun_all'] is defined) and (groups['zun_all
↪'] | length > 0) }}"

## Swift
horizon_swift_file_transfer_chunk_size: 524288

## Panel
horizon_default_panel: overview

# For blacklisting certain Nova extensions uncomment this configuration,
# and add necessary list items.
# horizon_nova_extensions_blacklist:
#   - "SimpleTenantUsage"

## Customization module
## See https://docs.openstack.org/horizon/latest/configuration/customizing.
↪html#horizon-customization-module-overrides
# horizon_customization_module: /local/path/to/customization_module.py

## Replace default theme files with your own
# horizon_custom_uploads:
#   logo:
#     src: "path_on_deployment_host_of_your_custom_logo.svg"
#     dest: "img/logo.svg"
#   logo_splash:
```

```yaml
#     src: "path_on_deployment_host_of_your_custom_logo-splash.svg"
#     dest: "img/logo-splash.svg"


_horizon_available_themes:
  default:
    theme_name: "default"
    theme_label: "Default"
    theme_path: "themes/default"
  material:
    theme_name: "material"
    theme_label: "Material"
    theme_path: "themes/material"

# Add custom themes. Deployers need to place the archive, with the theme␣
↪inside,
# into the directory, which is specified by theme_src_archive key.
# It should be an absolute path to the archive on the deployment host.
# Leading folders are not expected in the archive.
# Following archive formats are supported:
# .tar.gz, .tgz, .zip, .tar.bz, .tar.bz2, .tbz, .tbz2
# See https://docs.openstack.org/horizon/latest/configuration/themes.html
# for more details on custom themes
# Example:
#
# horizon_custom_themes:
#   custom_theme:
#     theme_name: "custom"
#     theme_label: "Custom"
#     theme_path: "themes/custom"
#     theme_src_archive: "/etc/openstack_deploy/horizon/themes/custom.tar.gz"
#
horizon_custom_themes: {}

# Which of the available themes will be used by default
# value is the theme_name from the vars above
horizon_default_theme: "default"

horizon_webroot: /

horizon_bind_address: "{{ openstack_service_bind_address | default('0.0.0.0')␣
↪}}"

horizon_listen_ports:
  http: "80"
  https: "443"

horizon_pip_packages:
  - "git+{{ horizon_git_repo }}@{{ horizon_git_install_branch }}#egg=horizon"
  - pymemcache
```

```yaml
  - python-memcached

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

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

# Optional pip packages for additional dashboards
# TODO(odyssey4me):
# Simplify this when we are no longer using the py_pkgs plugin
horizon_adjutant_optional_pip_packages:
  - "git+{{ adjutant_dashboard_git_repo }}@{{ adjutant_dashboard_git_install_
↪branch }}#egg=adjutant-ui"
horizon_barbican_optional_pip_packages:
  - "git+{{ barbican_dashboard_git_repo }}@{{ barbican_dashboard_git_install_
↪branch }}#egg=barbican-ui"
horizon_blazar_optional_pip_packages:
  - "git+{{ blazar_dashboard_git_repo }}@{{ blazar_dashboard_git_install_
↪branch }}#egg=blazar-dashboard"
horizon_cloudkitty_optional_pip_packages:
  - "git+{{ cloudkitty_dashboard_git_repo }}@{{ cloudkitty_dashboard_git_
↪install_branch }}#egg=cloudkitty_dashboard"
horizon_designate_optional_pip_packages:
  - "git+{{ designate_dashboard_git_repo }}@{{ designate_dashboard_git_
↪install_branch }}#egg=designate_dashboard"
horizon_heat_optional_pip_packages:
  - "git+{{ heat_dashboard_git_repo }}@{{ heat_dashboard_git_install_branch }}
↪#egg=heat_dashboard"
horizon_ironic_optional_pip_packages:
  - "git+{{ ironic_dashboard_git_repo }}@{{ ironic_dashboard_git_install_
↪branch }}#egg=ironic-ui"
horizon_magnum_optional_pip_packages:
  - "git+{{ magnum_dashboard_git_repo }}@{{ magnum_dashboard_git_install_
↪branch }}#egg=magnum-ui"
horizon_manila_optional_pip_packages:
  - "git+{{ manila_dashboard_git_repo }}@{{ manila_dashboard_git_install_
↪branch }}#egg=manila-ui"
horizon_masakari_optional_pip_packages:
  - "git+{{ masakari_dashboard_git_repo }}@{{ masakari_dashboard_git_install_
↪branch }}#egg=masakari_dashboard"
horizon_mistral_optional_pip_packages:
  - "git+{{ mistral_dashboard_git_repo }}@{{ mistral_dashboard_git_install_
↪branch }}#egg=mistral-dashboard"
horizon_murano_optional_pip_packages:
  - "git+{{ murano_dashboard_git_repo }}@{{ murano_dashboard_git_install_
↪branch }}#egg=murano-dashboard"
horizon_neutron_vpnaas_optional_pip_packages:
  - "git+{{ neutron_vpnaas_dashboard_git_repo }}@{{ neutron_vpnaas_dashboard_
```

```
↪git_install_branch }}#egg=neutron_vpnaas_dashboard"
horizon_octavia_optional_pip_packages:
  - "git+{{ octavia_dashboard_git_repo }}@{{ octavia_dashboard_git_install_
↪branch }}#egg=octavia_dashboard"
horizon_sahara_optional_pip_packages:
  - sahara_dashboard
horizon_tacker_optional_pip_packages:
  - "git+{{ tacker_dashboard_git_repo }}@{{ tacker_dashboard_git_install_
↪branch }}#egg=tacker_horizon"
horizon_trove_optional_pip_packages:
  - "git+{{ trove_dashboard_git_repo }}@{{ trove_dashboard_git_install_branch_
↪}}#egg=trove_dashboard"
horizon_watcher_optional_pip_packages:
  - "git+{{ watcher_dashboard_git_repo }}@{{ watcher_dashboard_git_install_
↪branch }}#egg=watcher_dashboard"
horizon_zun_optional_pip_packages:
  - "git+{{ zun_dashboard_git_repo }}@{{ zun_dashboard_git_install_branch }}
↪#egg=zun_ui"

# This variable is used to install additional pip packages
# that could be needed for additional dashboards required
# which are not part of the standard set above.
horizon_optional_pip_packages: []

# This variable is used to update the horizon translations from
# Zanata, this can be set to "True" when testing translations,
# but should otherwise be left as False.
horizon_translations_update: False

# This variable is used to define the version of the project
# (horizon) to pull from Zanata. Default value is master,
# other possibly values are stable/pike, stable/queens...
horizon_translations_project_version: "master"

# This variable is used to be able to fully override or
# partially union/intersect lists in order to change the
# behaviour of the pull of the translations per component.
horizon_translations_pull: "{{ _horizon_translations_pull }}"

# Set arbitrary horizon configuration options
horizon_config_overrides: {}

# Set overrides for horizon embedded policies
#horizon_policy_overrides:
#  cinder:
#    "volume:create": "rule:admin_or_owner"
horizon_policy_overrides: {}

horizon_init_overrides: {}
```

```
horizon_uwsgi_ini_overrides: {}

horizon_keystone_admin_roles:
  - admin

# Set the "credentials" authentication choice to show as default.
# The list of authentication mechanisms which include keystone
# federation protocols and identity provider/federation protocol
#horizon_websso_keystone_url: "{{ horizon_keystone_endpoint }}"
horizon_websso_initial_choice: "credentials"
horizon_websso_default_redirect: False
horizon_websso_default_redirect_region: "{{ horizon_websso_keystone_url |
↪default(horizon_keystone_endpoint) }}"
horizon_websso_default_redirect_logout: ""
horizon_websso_use_http_referer: True
```

DEPENDENCIES

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

To use this role, define the following variables:

```yaml
horizon_galera_address: 10.100.100.101
horizon_container_mysql_password: "SuperSecrete"
horizon_secret_key: "SuperSecreteHorizonKey"
```

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

EXAMPLE PLAYBOOK

```yaml
---
- name: Installation and setup of horizon
  hosts: horizon_all
  user: root
  roles:
    - { role: "os_horizon", tags: [ "os-horizon" ] }
  vars:
    galera_client_drop_config_file: false
    external_lb_vip_address: 10.100.100.101
    internal_lb_vip_address: 10.100.100.101
    horizon_galera_address: 10.100.100.101
    horizon_container_mysql_password: "SuperSecrete"
    horizon_secret_key: "SuperSecreteHorizonKey"
    horizon_external_ssl: true
    rabbitmq_servers: 10.100.100.101
    rabbitmq_use_ssl: false
    rabbitmq_port: 5671
    keystone_admin_user_name: admin
    keystone_auth_admin_password: "SuperSecretePassword"
    keystone_admin_tenant_name: admin
    keystone_service_adminuri_insecure: false
    keystone_service_internaluri_insecure: false
    keystone_service_internaluri: "http://{{ internal_lb_vip_address }}:5000"
    keystone_service_internalurl: "{{ keystone_service_internaluri }}/v3"
    keystone_service_adminuri: "http://{{ internal_lb_vip_address }}:5000"
    keystone_service_adminurl: "{{ keystone_service_adminuri }}/v3"
    openrc_os_password: "{{ keystone_auth_admin_password }}"
    openrc_os_domain_name: "Default"
    memcached_servers: 10.100.100.101
    memcached_encryption_key: "secrete"
    galera_root_user: root
  vars_prompt:
    - name: "galera_root_password"
      prompt: "What is galera_root_password?"
```

4.1 Tags

This role supports two tags: `horizon-install` and `horizon-config`.

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

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