
OpenStack-Ansible Documentation:

repo_server role

Release 18.1.0.dev164

OpenStack-Ansible Contributors

Jan 23, 2025

CONTENTS

1	Default variables	3
2	Required variables	7
3	Example playbook	9

Ansible role that deploys a repository server for python packages, git sources and package caching for deb/rpm.

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

DEFAULT VARIABLES

```
## Verbosity Options
debug: False

## Cap the maximum number of threads / workers when a user value is
↳ unspecified.
repo_nginx_threads_max: 16
repo_nginx_threads: "{{ [[ansible_facts['processor_vcpus']]|default(2) // 2,
↳ 1] | max, repo_nginx_threads_max] | min }}"

## APT Cache Options
cache_timeout: 600

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

repo_worker_connections: 1024
repo_server_name: openstack-slushee

repo_service_home_folder: /var/www
repo_service_user_name: nginx
repo_service_group_name: www-data

# Main web server port
repo_server_bind_address: "{{ openstack_service_bind_address | default('0.0.0.
↳ 0') }}"
repo_server_port: 8181

# This directory is used by the repo_build, and will cause problems if synced
# to repo_containers with other releases.
repo_build_global_links_dirname: links

# This directory is used on the deploy host to create u-c files which are then
# copied to the repo server and served by http. Any other files in this
# directory placed by the deployer will also be transferred
repo_upper_constraints_path: "/etc/openstack_deploy/upper-constraints"
```

(continues on next page)

(continued from previous page)

```

# Multiple repo servers must have a shared /var/www/repo
repo_server_systemd_mounts: []

# Example using remote shared filesystem to synchronise the repo contents,
↳between
# several repo servers
#repo_server_systemd_mounts:
# - what: "gluster-server:gluster-volume-name"
#   where: "/var/www/repo"
#   type: glusterfs
#   state: 'started'
#   enabled: true

###
### Backend TLS
###

# Define if communication between haproxy and service backends should be
# encrypted with TLS.
repo_backend_ssl: "{{ openstack_service_backend_ssl | default(False) }}"

# Storage location for SSL certificate authority
repo_pki_dir: "{{ openstack_pki_dir | default('/etc/openstack_deploy/pki') }}"

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

# repo server certificate
repo_pki_keys_path: "{{ repo_pki_dir ~ '/certs/private/' }}"
repo_pki_certs_path: "{{ repo_pki_dir ~ '/certs/certs/' }}"
repo_pki_intermediate_cert_name: "{{ openstack_pki_service_intermediate_cert_
↳name | default('ExampleCorpIntermediate') }}"
repo_pki_intermediate_cert_path: "{{ repo_pki_dir ~ '/roots/' ~ repo_pki_
↳intermediate_cert_name ~ '/certs/' ~ repo_pki_intermediate_cert_name ~ '.crt
↳' }}"
repo_pki_san: "{{ openstack_pki_san | default('DNS:' ~ ansible_facts['hostname
↳'] ~ ',IP:' ~ management_address) }}"
repo_pki_regen_cert: ''
repo_pki_certificates:
  - name: "repo-{{ ansible_facts['hostname'] }}"
    provider: ownca
    cn: "{{ ansible_facts['hostname'] }}"
    san: "{{ repo_pki_san }}"
    signed_by: "{{ repo_pki_intermediate_cert_name }}"

# repo destination files for SSL certificates
repo_ssl_cert: /etc/ssl/certs/repo.pem
repo_ssl_key: /etc/ssl/private/repo.key
repo_ssl_ca_cert: /etc/ssl/certs/repo-ca.pem

```

(continues on next page)

(continued from previous page)

```
# Installation details for SSL certificates
repo_pki_install_certificates:
  - src: "{{ repo_user_ssl_cert | default(repo_pki_certs_path ~ 'repo_' ~
↪ansible_facts['hostname'] ~ '-chain.crt') }}"
    dest: "{{ repo_ssl_cert }}"
    owner: "{{ repo_service_user_name }}"
    group: "{{ repo_service_group_name }}"
    mode: "0644"
  - src: "{{ repo_user_ssl_key | default(repo_pki_keys_path ~ 'repo_' ~
↪ansible_facts['hostname'] ~ '.key.pem') }}"
    dest: "{{ repo_ssl_key }}"
    owner: "{{ repo_service_user_name }}"
    group: "{{ repo_service_group_name }}"
    mode: "0600"
  - src: "{{ repo_user_ssl_ca_cert | default(repo_pki_intermediate_cert_path) ~
↪ }}"
    dest: "{{ repo_ssl_ca_cert }}"
    owner: "{{ repo_service_user_name }}"
    group: "{{ repo_service_group_name }}"
    mode: "0644"
    condition: "{{ repo_user_ssl_ca_cert is defined }}"

# Define user-provided SSL certificates
#repo_user_ssl_cert: <path to cert on ansible deployment host>
#repo_user_ssl_key: <path to cert on ansible deployment host>
#repo_user_ssl_ca_cert: <path to cert on ansible deployment host>
```


REQUIRED VARIABLES

None.

EXAMPLE PLAYBOOK

```
- name: Setup repo servers
  hosts: repo_all
  user: root
  roles:
    - { role: "repo_server", tags: [ "repo-server" ] }
```