OpenStack-Ansible Documentation: memcached_server role

Release 18.1.0.dev86

OpenStack-Ansible Contributors

Dec 18, 2020

CONTENTS

CHAPTER

ONE

MAKING MEMCACHED HIGHLY-AVAILABLE

By default Memcached servers are deployed on each controller host as a part of *shared-infra_containers* group. Drivers, like oslo_cache.memcache_pool support marking memcache backends as dead, however not all services allow you to select driver which will be used for interaction with Memcached. In the meanwhile you may face services API response delays or even unresponsive APIs while one of the memcached backends is down. Thats why you may want to use HAProxy for handling access and check of backend aliveness.

1.1 Configuring Memcached through HAProxy

Setting haproxy in front of the Memcached servers and relying it in checking aliveness of the backends gives more reliable failover and minimize delays in case of backend failure. We need to define the following in your `user_variables.yml`:

```
memcached_servers: "{{ internal_lb_vip_address ~ ':' ~ memcached_port }}"
haproxy_extra_services:
  haproxy_service_name: memcached
  haproxy_backend_nodes: "{{ groups['memcached'] | default([]) }}"
  haproxy_bind: "{{ [internal_lb_vip_address] }}"
  haproxy_port: 11211
  haproxy_balance_type: tcp
  haproxy_balance_alg: source
  haproxy_backend_ssl: False
  haproxy_backend_options:
    - tcp-check
  haproxy_whitelist_networks: "{{ haproxy_memcached_whitelist_networks }}"
```

After setting that you need to update haproxy and all services configuration to use new memcached backend:

```
# openstack-ansible playbooks/haproxy-install.yml
# openstack-ansible playbooks/setup-openstack.yml
```

Ansible role to install and configure Memcached.

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

CHAPTER

TWO

DEFAULT VARIABLES

```yaml
## Logging level
debug: False

## APT Cache Options
cache_timeout: 600

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

# MemcacheD could set 'PrivateDevices=True' for its systemd unit by␣
↪default when
# installed into a container. This provides some additional security, but␣
↪it
# causes problems with creating mount namespaces on CentOS 7 with systemd␣
↪219.
# While the security enhancements are helpful on bare metal hosts with
# multiple services running, they are not as helpful when MemcacheD is␣
↪running
# in a container with its own isolated namespaces.
#
# Related bugs:
#   https://bugs.launchpad.net/openstack-ansible/+bug/1697531
#   https://github.com/lxc/lxc/issues/1623
#   https://github.com/systemd/systemd/issues/6121
#
# Setting the following variable to 'yes' will disable the PrivateDevices
# setting in the systemd unit file for MemcacheD on CentOS 7 hosts.
memcached_disable_privatedevices: "{{ ansible_pkg_mgr == 'yum' }}"

# The default memcache memory setting is to use .25 of the available␣
↪system ram
# as long as that value is < 8192. However you can set the `memcached_
↪memory`
# value to whatever you like as an override.
base_memcached_memory: "{{ ansible_memtotal_mb | default(4096) }}"
memcached_memory: "{{ base_memcached_memory | int // 4 if base_memcached_
↪memory | int // 4 < 8192 else 8192 }}"

memcached_port: 11211
memcached_listen: "127.0.0.1"
memcached_connections: 4096
memcached_threads: 4
```

(continues on next page)

(continued from previous page)

```
memcached_file_limits: "{{ memcached_connections | int + 1024 }}"

install_test_packages: False
```

CHAPTER

THREE

REQUIRED VARIABLES

None

CHAPTER

FOUR

EXAMPLE PLAYBOOK

```
- name: Install Memcached
  hosts: memcached
  user: root
  roles:
    - { role: "memcached_server" }
```