```
##############################################################################
# Copyright 2015 F5 Networks Inc.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
##############################################################################
#
#                    ############
#                  ################
#                ###/ _ \###|     |#
#               ###| |#| |##| |######
#              ####| |######| |######
#              ##|     |####\    \###    AGILITY YOUR WAY!
#              ####| |##########| |###
#              ####| |##########| |##
#               ###| |########/ /##
#                #|    |####|  /##
#                 ###############
#                  ############
#
#                   NETWORKS
#
##############################################################################
#
[DEFAULT]
# Show debugging output in log (sets DEBUG log level output).
debug = True
# The LBaaS agent will resync its state with Neutron to recover from any
# transient notification or rpc errors. The interval is number of
# seconds between attempts.
#
periodic_interval = 10
#
# How often should the agent throw away its service cache and
# resync assigned services with the neutron LBaaS plugin.
#
# service_resync_interval = 500
#
# Objects created on the BIG-IP by this agent will have their names prefixed
# by an environment string. This allows you set this string.  The default is
# 'uuid'.
#
# WARNING - you should only set this before creating any objects.  If you change
# it with established objects, the objects created with an alternative prefix,
# will no longer be associated with this agent and all objects in neutron
# and on the the BIG-IP associated with the old environment will need to be managed
# manually.
#
# environment_prefix = uuid
#
# Static configuration data to send back to the plugin. This can be used
# on the plugin side of neutron to provide agent identification for custom
# pool to agent scheduling. This should be a single or comma separated list
# of name:value entries which will be sent in the agent's configuration
# dictionary to neutron.
# static_agent_configuration_data = name1:value1, name1:value2, name3:value3
#
##############################################################################
#  Device Setting
##############################################################################
#
```

```
# Device type for LBaaS: valid type are:
#
#   external - external  (hardware of VE)
#   guest_admin - VE created under the admin tenant
#   guest_tenant - VE created under the pool tenant
#
f5_device_type = external
#
# HA model
#
# Device can be required to be:
#
# standalone - single device no HA
# pair - active/standby two device HA
# scalen - active device cluster
#
# If the device is external, the devices must be onboarded for the
# appropriate HA mode or else the driver will not provision devices
#
f5_ha_type = pair
#
#
# Sync mode
#
# autosync - syncable policies configured on one device then
#            synced to the group
# replication - each device configured separately
#
f5_sync_mode = replication
#

################################################################################
#  L2 Segmentation Mode Settings
################################################################################
#
# Device VLAN to interface and tag mapping
#
# For pools or VIPs created on networks with type VLAN we will map
# the VLAN to a particular interface and state if the VLAN tagging
# should be enforced by the external device or not.  This setting
# is a comma separated list of the following format:
#
#    physical_network:interface_name:tagged, physical:interface_name:tagged
#
# where :
#   physical_network corresponds to provider:physical_network attributes
#   interface_name is the name of an interface or LAG trunk
#   tagged is a boolean (True or False)
#
# If a network does not have a provider:physical_network attribute,
# or the provider:physical_network attribute does not match in the
# configured list, the 'default' physical_network setting will be
# applied. At a minimum you must have a 'default' physical_network
# setting.
#
# standalone example:
#   f5_external_physical_mappings = default:1.1:True
#
# pair or scalen (1.1 and 1.2 are used for HA purposes):
#   f5_external_physical_mappings = default:1.3:True
#
f5_external_physical_mappings = default:1.3:True
#
# Device Tunneling (VTEP) selfips
#
# This is a single entry or comma separated list of cidr (h/m) format
# selfip addresses, one per BIG-IP device, to use for VTEP addresses.
#
# If no gre or vxlan tunneling is required, these settings should be
# commented out or set to None.
#
```

```
f5_vtep_folder = 'Common'
f5_vtep_selfip_name = 'vtep'
#
#
# Tunnel types
#
# This is a comma separated list of tunnel types to report
# as available from this agent as well as to send via tunnel_sync
# rpc messages to compute nodes. This should match your ml2
# network types on your compute nodes.
#
# If you are using only gre tunnels it should be:
#
# advertised_tunnel_types = gre
#
# If you are using only vxlan tunnels it should be:
#
# advertised_tunnel_types = vxlan
#
# If this agent could get both gre and vxlan tunnel networks:
#
# advertised_tunnel_types = gre,vxlan
#
# If you are using only vlans only it should be:
#
# advertised_tunnel_types =
#
# Static ARP population for members on tunnel networks
#
# This is a boolean True or False value which specifies
# that if a Pool Member IP address is associated with a gre
# or vxlan tunnel network, in addition to a tunnel fdb
# record being added, that a static arp entry will be created to
# avoid the need to learn the member's MAC address via flooding.
#
# f5_populate_static_arp = True
#
# Device Tunneling (VTEP) selfips
#
# This is a boolean entry which determines if they BIG-IP will use
# L2 Population service to update its fdb tunnel entries. This needs
# to be setup in accordance with the way the other tunnel agents are
# setup.  If the BIG-IP agent and other tunnel agents don't match
# the tunnel setup will not work properly.
#
l2_population = True


##############################################################################
#  L3 Segmentation Mode Settings
##############################################################################
#
# Global Routing Mode - No L2 or L3 Segmentation on BIG-IP
#
# This setting will cause the agent to assume that all VIPs
# and pool members will be reachable via global device
# L3 routes, which must be already provisioned on the BIG-IPs.
#
# In f5_global_routed_mode, BIG-IP will not assume L2
# adjacentcy to any neutron network, therefore no
# L2 segementation between tenant services in the data plane
# will be provisioned by the agent. Because the routing
# is global, no L3 SelfIPs or SNATs will be provisioned
# by the agent on behalf of tenants either. You must have
# all necessary L3 routes (including TMM default routes)
# provisioned before LBaaS resources are provisioned for tenants.
#
# WARNING: setting this mode to True will override
# the use_namespaces, setting it to False, because only
# one global routing space will used on the BIG-IP.  This
# means overlapping IP addresses between tenants is no
```

```
# longer supported.
#
# WARNING: setting this mode to True will override
# the f5_snat_mode, setting it to True, because pool members
# will never be considered L2 adjacent to the BIG-IP by
# the agent. All member access will be via L3 routing, which
# will need to be setup on the BIG-IP before LBaaS provisions
# resources on behalf of tenants.
#
# WARNING: setting this mode to True will override the
# f5_snat_addresses_per_subnet, setting it to 0 (zero).
# This will force all VIPs to use AutoMap SNAT for which
# enough SelfIP will need to be pre-provisioned on the
# BIG-IP to handle all pool member connections. The SNAT,
# an L3 mechanism, will all be global without reference
# to any specific tenant SNAT pool.
#
# WARNIG: setting this mode will make the VIPs listen
# on all provisioned L2 segments (All VLANs). This is
# because no L2 information will be taken from
# neutron, thus making the assumption that all VIP
# L3 addresses will be globally routable without
# segmentation at L2 on the BIG-IP.
#
f5_global_routed_mode = False
#
# Allow overlapping IP for tenant networks.
#
# This setting is forced to False if
# f5_global_routed_mode = True.
#
use_namespaces = True
#
# Dictates the strict isolation of the routing
# tables.  If you set this to True, then all
# VIPs and Members must be in the same tenant
# or less they can't communicate.
#
# This setting is only valid if use_namespaces = True.
#
f5_route_domain_strictness = False
#
# SNAT Mode and SNAT Address Counts
#
# This setting will force the use of SNATs.
#
# If this is set to False, a SNAT will not
# be created (routed mode) and the BigIP
# will attempt to setup a floating SelfIP
# as the subnet's default gateway address.
# and a wild card IP forwarding virtual
# server will be setup on member's network.
# Setting this to False will mean Neutron
# Floating Self IPs will not longer work
# if the same BigIP device is not being used
# as the Neutron Router implementation.
#
# This setting will be forced to True if
# f5_global_routed_mode = True.
#
f5_snat_mode = True
#
# This setting will specify the number of snat
# addresses to put in a snat pool for each
# subnet associated with a created local Self IP.
#
# Setting to 0 (zero) will set VIPs to AutoMap
# SNAT and the device's local Self IP will
# be used to SNAT traffic.
#
# In scalen HA mode, this is the number of snat
```

```
# addresses per active traffic-group at the time
# a service is provisioned.
#
# This setting will be forced to 0 (zero) if
# f5_global_routed_mode = True.
#
f5_snat_addresses_per_subnet = 1
#
# This setting will cause all networks with
# the router:external attribute set to True
# to be created in the Common partition and
# placed in route domain 0.
f5_common_external_networks = True
#
#
# Common Networks
#
# This setting contains a name value pair comma
# separated list where if the name is a neutron
# network id used for a vip or a pool member,
# the network should not be created or deleted
# on the BIG-IP, but rather assumed that the value
# is the name of the network already created in
# the Common partition with all L3 addresses
# assigned to route domain 0.  This is useful
# for shared networks which are already defined
# on the BIG-IP prior to LBaaS configuration. The
# network should not be managed by the LBaaS agent,
# but can be used for VIPs or pool members
#
# If your Internet VLAN on your BIG-IP is named
# /Common/external, and that corresponds to
# Neutron uuid: 71718972-78e2-449e-bb56-ce47cc9d2680
# then the entry would look like:
#
# common_network_ids = 71718972-78e2-449e-bb56-ce47cc9d2680:external
#
# If you had multiple common networks, they are simply
# comma seprated like this example:
#
# common_network_ids = 71718972-78e2-449e-bb56-ce47cc9d2680:external,396e06a0-05c7-4a49-8e86-
04bb83d14438:vlan1222
#
# The default is no common networks defined
#
# L3 Bindings
#
# Some systems require the need to bind L3 addresses
# to specific ports, often for security.
#
# An example would be if a LBaaS iControl endpoint is using
# untagged VLANs and is a nova guest instance. By
# default, neutron will attempt to apply security rule
# for anti-spoofing which will not allow just any L3
# address to be used on the neutron port. The answer is to
# use allowed-address-pairs for the neutron port.
#
# What is required is a software hook which allows the binding.
# l3_binding_driver needs to reference a subclass of the L3BindingBase
# class and provides the methods to bind and unbind L3 address
# to ports.
#
# l3_binding_driver =
neutron.services.loadbalancer.drivers.f5.bigip.l3_binding.AllowedAddressPairs
#
# The l3_binding_static_mappings allows for a JSON encoded dictionary
# mapping neutron subnet ids to lists of L2 ports and devices which
# require mapping. The entries for port and device mappings
# vary between providers. They may look like a neutron port id
# and a nova guest instance id.
#
```

```
# In addition to any static mappings, when the iControl endpoints
# are initialized, all their TMM MAC addresses will be collect
# and neutron will be queried to see if they MAC addresses
# correspond to known neutron ports. If they do, automatic entries
# for all mapped fixed_ips will be made referencing the ports id
# and the ports device_id.
#
# l3_binding_static_mappings = 'subnet_a':[('port_a','device_a'),('port_b','device_b')],
'subnet_b':[('port_c','device_a'),('port_d','device_b')]
#
#

################################################################################
#  Device Driver Setting
################################################################################
#
f5_bigip_lbaas_device_driver =
neutron.services.loadbalancer.drivers.f5.bigip.icontrol_driver.iControlDriver
#


################################################################################
#  Device Driver - iControl Driver Setting
################################################################################
#
# icontrol_hostname is valid for external device type only.
# this setting can be either a single IP address or a
# comma separated list contain all devices in a device
# service group.  For guest devices, the first fixed_address
# on the first device interfaces will be used.
#
# If a single IP address is used and the HA model
# is not standalone, all devices in the sync failover
# device group for the hostname specified must have
# there management IP address reachable to the agent.
# If order to access devices' iControl interfaces via
# SelfIPs, you should specify them as a comma
# separated list below.
#
icontrol_hostname = 192.168.1.245

# If you are using VCMP with VLANs, you will need to configure
# your vcmp host addresses, in addition to the guests addresses.
# VCMP Host access is necessary for provisioning VLANs to a guest.
# Use icontrol_hostname for VCMP guests and icontrol_vcmp_hostname
# for VCMP hosts. The plug-in will automatically determine
# which host corresponds to each guest.
icontrol_vcmp_hostname = 192.168.1.245

#
# icontrol_username must be a valid Administrator username
# on all devices in a device sync failover group.
#
icontrol_username = admin
#
# icontrol_password must be a valid Administrator password
# on all devices in a device sync failover group.
#
icontrol_password = admin
#
icontrol_connection_retry_interval = 10
#
icontrol_connection_timeout = 10

# icontrol_config_mode = [ iapp | object ]
# defaults to iapp based config
icontrol_config_mode = iapp

# BIG-IQ configuration
bigiq_hostname =
```

```
bigiq_admin_password =

# These openstack credentials are used when BIG-IQ is configured
# in order to determine whether the tenant has a BIG-IP that
# the BIG-IQ can discover.
openstack_keystone_uri =

openstack_admin_username =

openstack_admin_password =

# These are the default credentials used by BIG-IQ to manage BIG-IPs
bigip_management_username =

bigip_management_password =
```