

Draft

# A10Networks LBaaS Driver for Thunder

---

Mike Thompson

Draft

Draft

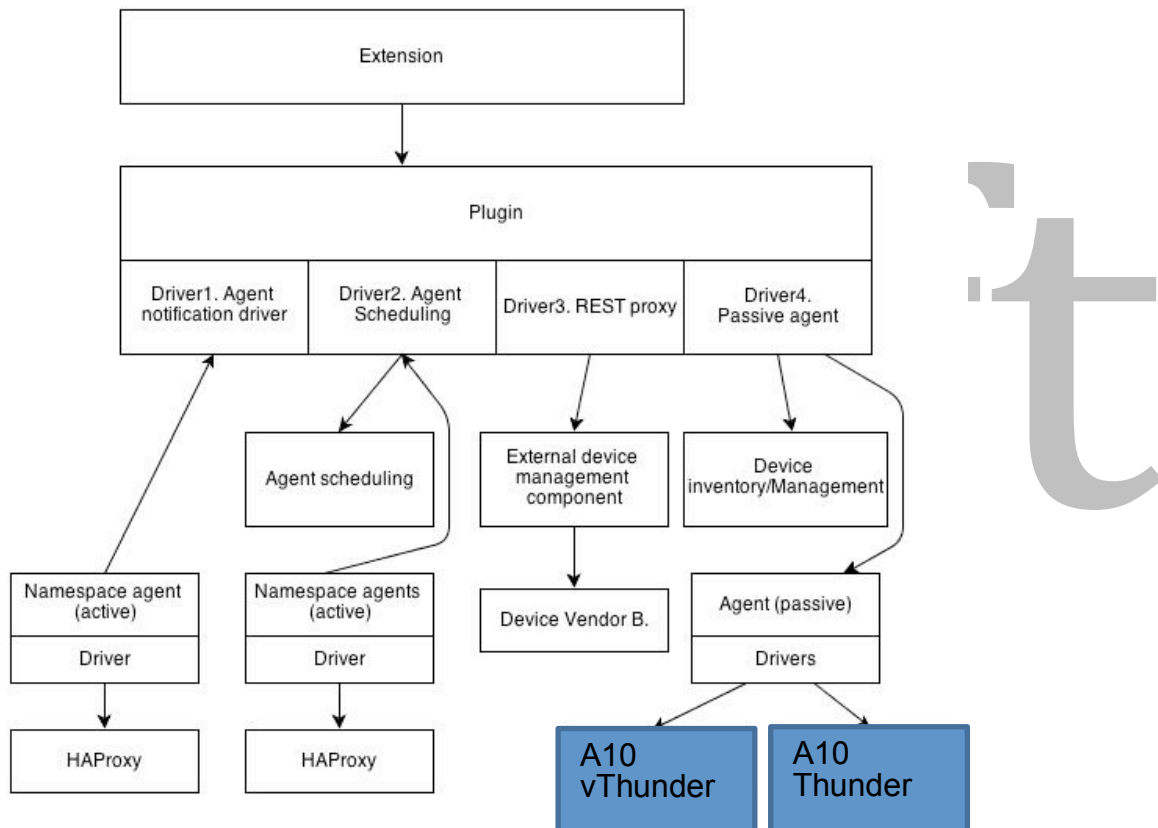
## Introductions:

This A10 Network Driver is specifically designed to manage Thunder Series Appliances. There are two ways that a device can register a device.

1. Create different versions of the driver and associated configuration file.
2. Utilize, the first method, but also utilize multiple devices in the configuration file.

\*\* A non-community version of the driver can be provided which extends horizon and other component.

## Implementation:



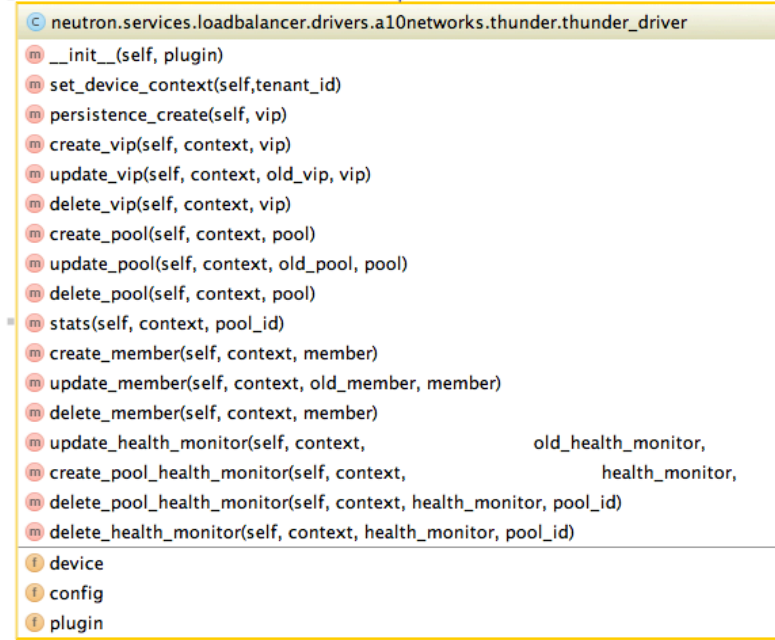
The implementation relies on:

Added A10 LBaaS Driver to branch.  
new file: a10networks/\_\_init\_\_.py

Draft

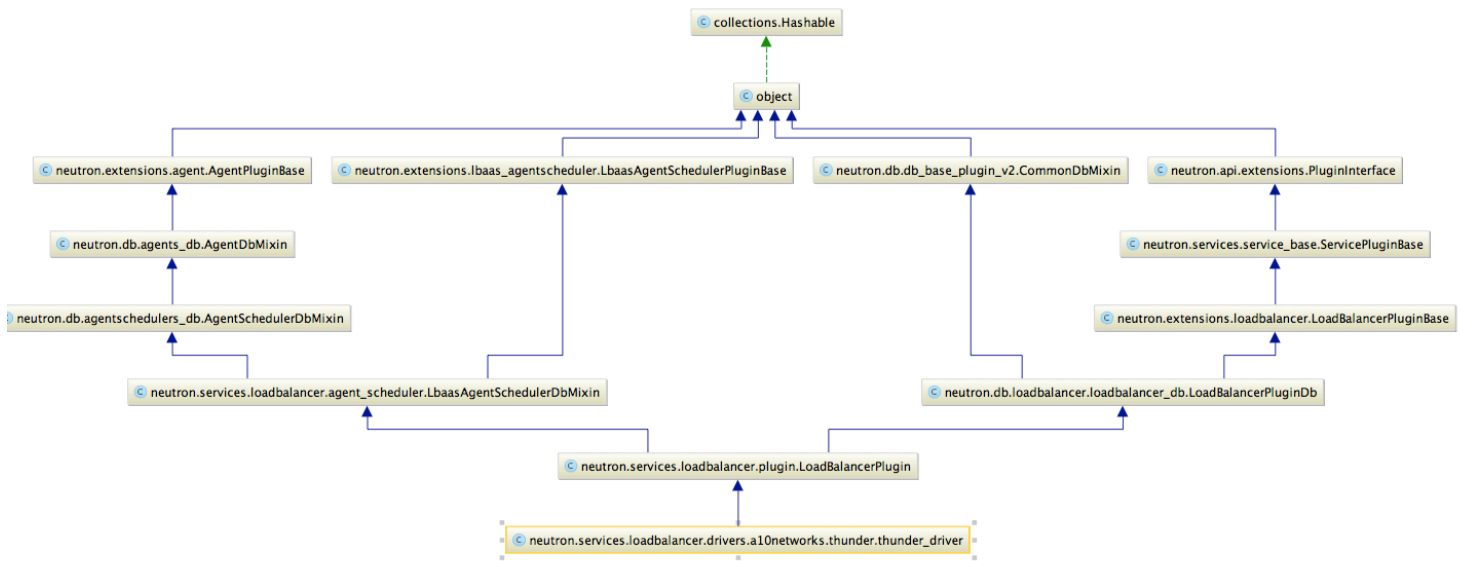
new file: a10networks/a10\_api\_2\_1/\_init\_.py  
new file: a10networks/a10\_api\_2\_1/core/\_init\_.py  
new file: a10networks/a10\_api\_2\_1/core/a10logger.py  
new file: a10networks/a10\_api\_2\_1/core/apiObj.py  
new file: a10networks/a10\_api\_2\_1/core/device\_select.py  
new file: a10networks/a10\_api\_2\_1/core/ssl\_monkeypatch.py  
new file: a10networks/a10\_api\_2\_1/core/structure.py  
new file: a10networks/a10\_api\_2\_1/slb/\_init\_.py  
new file: a10networks/a10\_api\_2\_1/slb/hm.py  
new file: a10networks/a10\_api\_2\_1/slb/server.py  
new file: a10networks/a10\_api\_2\_1/slb/service\_group.py  
new file: a10networks/a10\_api\_2\_1/slb/ssl.py  
new file: a10networks/a10\_api\_2\_1/slb/template.py  
new file: a10networks/a10\_api\_2\_1/slb/virtual\_server.py  
new file: a10networks/a10\_api\_2\_1/slb/virtual\_service.py  
new file: a10networks/device\_conf.ini  
new file: a10networks/exceptions.py  
new file: a10networks/thunder.py

### Implemented methods in driver:



```
neutron.services.loadbalancer.drivers.a10networks.thunder.thunder_driver
  m __init__(self, plugin)
  m set_device_context(self,tenant_id)
  m persistence_create(self, vip)
  m create_vip(self, context, vip)
  m update_vip(self, context, old_vip, vip)
  m delete_vip(self, context, vip)
  m create_pool(self, context, pool)
  m update_pool(self, context, old_pool, pool)
  m delete_pool(self, context, pool)
  m stats(self, context, pool_id)
  m create_member(self, context, member)
  m update_member(self, context, old_member, member)
  m delete_member(self, context, member)
  m update_health_monitor(self, context, old_health_monitor, health_monitor)
  m create_pool_health_monitor(self, context, health_monitor, pool_id)
  m delete_pool_health_monitor(self, context, health_monitor, pool_id)
  m delete_health_monitor(self, context, health_monitor, pool_id)
  f device
  f config
  f plugin
```

Draft



## Neutron.conf

In the Neutron.conf add the following line. This will enable the specific service provider. You can register multiple drivers that point to a specific system or group of systems.

```
service_provider
=LOADBALANCER:A10_Thunder:neutron.services.loadbalancer.drivers.a10networks.thunder.thunde
r_driver:default
```

Draft

### device\_config.ini

Can utilize multiple devices or one device.

[devices]

#format is device FQDN then host object

```
#ax1.foo.com = {"host":"192.168.212.120",
```

```
#     "username" : "admin",
```

```
#     "password" : "a10",
```

```
#     "status":"1",
```

```
#     "autosnat":True}
```

```
ax1.foo.com = {"host":"192.168.212.120",
```

```
    "username":"admin",
```

```
    "password":"a10", "status":"1", "autosnat":"True"}
```

[placement]

#Only method supported is hash

#method=hash

[member]

#This is for the nat address on router.

#If set to false then will utilize default member addr.

use\_float\_if\_assigned = 'True'

[partition]

#Choices in this version(ADP)

#Choices in the next release(L3V, LSI, VSI)

#partition='ADP'

### Reference Architecture:

Please contact a10 networks for more information.

1. One arm mode with SNAT.
2. Routed Mode on Private network
3. Route Injection Mode.