gnt-network design improvements

Dimitris Aragiorgis

Greek Research & Technology Network

dimara@grnet.gr

September 4, 2013

Overview

1. Before gnt-network
2. Synnefo usecase
3. Future Work
    - Extend external scripts
    - Abstract Networks
    - nicparams inheritance
    - gnt-network + OVS

Table of Contents

1. Before gnt-network
2. Synnefo usecase
3. Future Work
   - Extend external scripts
   - Abstract Networks
   - nicparams inheritance
   - gnt-network + OVS

MAC + IP + link + mode = NOT enough

Limited NIC configuration options

- No subnet provided (e.g. DHCP response)
- ...

MAC + IP + link + mode = NOT enough

Limited NIC configuration options

- No subnet provided (e.g. DHCP response)
- ...

Poor Management

- A VM wants an IP. Which is available? Try and error?
- ...

Table of Contents

1. Before gnt-network

2. Synnefo usecase

3. Future Work
   - Extend external scripts
   - Abstract Networks
   - nicparams inheritance
   - gnt-network + OVS

Ensure isolation

- routed mode with proxy arp (`ip route`, `ip rule`, `arptables`)
- private networks over physical vlans (`vconfig`)
- private networks over common bridge (`ebtables`)

Networking in the Cloud: Ganeti + Synnefo

Ensure isolation

- routed mode with proxy arp (`ip route, ip rule, arptables`)
- private networks over physical vlans (`vconfig`)
- private networks over common bridge (`ebtables`)

Ensure connectivity

- custom kvm-ifup/vif-ganeti scripts [snf-network]
- node level dhcpd based on NFQUEUE [nfdhcpd]
- update external dns server with custom ganeti hook

IP allocation in the Cloud

Clusters	Resource	Ganeti	Synnefo
One	Exclusive	X	
Many	Exclusive	X	
Many	Shared		X

Challenges

- easy way to assign IPs to instances [1] [2]
- provide a way to configure each NIC differently
- find a way to hide underlying infrastructure
- better networking overview

[1] In multiple clusters with shared IPs, allocation must be done externally.

[2] Still Ganeti could double check for cluster wide uniqueness.

Current gnt-network support

- Provides an IP pool
  `gnt-network add --network 192.168.1.0/24 net1`
- Abstracts network infra
  `gnt-network connect net1 bridged prv0`
- Supports network tags
  `nfdhcpd, mac-filtered, ip-less-routed, physical-vlan`
- Assigns IPv6 prefix and gateway per network
  `--network6, --gateway6`

Current gnt-network support

- Provides an IP pool
  `gnt-network add --network 192.168.1.0/24 net1`
- Abstracts network infra
  `gnt-network connect net1 bridged prv0`
- Supports network tags
  `nfdhcpd, mac-filtered, ip-less-routed, physical-vlan`
- Assigns IPv6 prefix and gateway per network
  `--network6, --gateway6`

gnt-network *alone* does **not** ensure connectivity

Table of Contents

1. Before gnt-network
2. Synnefo usecase
3. Future Work
   - Extend external scripts
   - Abstract Networks
   - nicparams inheritance
   - gnt-network + OVS

kvm-ifup, vif-ganeti

Currently act depending on NIC's mode: `bridged`, `routed`, `ovs`

kvm-ifup, vif-ganeti

Currently act depending on NIC's mode: `bridged, routed, ovs`

Missing:

1. make use of NIC's network info
2. apply corresponding rules depending on network tags
3. update external dhcpd entries (e.g. create [nfdhcpd] binding files)
4. provide hook to update dns entries

kvm-ifup, vif-ganeti

Currently act depending on NIC's mode: `bridged, routed, ovs`

Missing:

1. make use of NIC's network info
2. apply corresponding rules depending on network tags
3. update external dhcpd entries (e.g. create [nfdhcpd] binding files)
4. provide hook to update dns entries

Why not use [snf-network] as default?

Abstract Networks

Current state: Ganeti Network ⇔ netparams + IP pool

Abstract Networks

Current state: Ganeti Network ⇔ netparams + IP pool

Generic info

1. name
2. tags

Abstract Networks

Current state: Ganeti Network ⇔ netparams + IP pool

Generic info

1. name
2. tags

L2 = Collision domain

1. mode
2. link
3. vlan
4. MAC prefix

Abstract Networks

Current state: Ganeti Network ⇔ netparams + IP pool

Generic info

1. name
2. tags

L3 = TCP/IP stuff

1. IPv4 subnet/gateway
2. IPv6 prefix/gateway

L2 = Collision domain

1. mode
2. link
3. vlan
4. MAC prefix

Abstract Networks

Current state: Ganeti Network ⇔ netparams + IP pool

Generic info

1. name
2. tags

L3 = TCP/IP stuff

1. IPv4 subnet/gateway
2. IPv6 prefix/gateway

L2 = Collision domain

1. mode
2. link
3. vlan
4. MAC prefix

IP pool

- make it optional
- no need to burden config.data in case allocation is done externally (multiple ganeti clusters)

nicparams vs netparams

Current Implementation

- nicparams are hardcoded inside NIC objects
- Are evaluated once NIC is created or modified
- In case of a NIC is attached to a network they are inherited from nodegroup's netparams.

nicparams vs netparams

Current Implementation

- nicparams are hardcoded inside NIC objects
- Are evaluated once NIC is created or modified
- In case of a NIC is attached to a network they are inherited from nodegroup's netparams.

Proposed Implementation

Evaluate NIC params on the fly in case NIC resides in a network.
Change collision domain **only** by:
a) reconnecting network and b) rebooting instances.

gnt-network + OVS

Once abstract networks are implemented (and L2 gets separated from L3):

1. create an L2 network
2. connect it to the nodegroup with desired netparams (mode, link, vlan)
3. setup node level OVS configuration via RPC or hooks (currently done via `gnt-node add`)

References

- snf-network (0.14.0)
  https://code.grnet.gr/git/snf-network
  deb http://apt.dev.grnet.gr/ squeeze/.
- nfdhcpd (0.11.5-2)
  https://code.grnet.gr/git/snf-nfdhcpd
  deb http://apt.dev.grnet.gr/ squeeze/.

Thanks!