

oVirt

Using NetworkManager for host networking in oVirt 4.4

Ales Musil
Software Engineer

September/2020



Agenda

- Motivation
- What is nmstate?
- Architecture
- Troubleshooting
- Implications

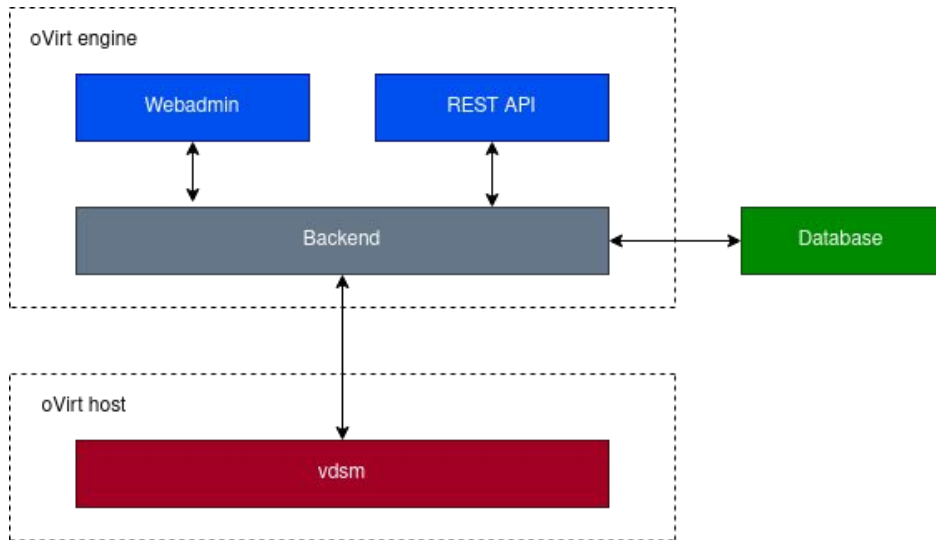
Motivation

- Common building blocks with OpenShift
 - KubeVirt is using nmstate already
 - Kubernetes-nmstate
 - Enables the management of network configuration of bare-metal Kubernetes hosts
- Previous network-scripts based approach is deprecated in CentOS 8

nmstate

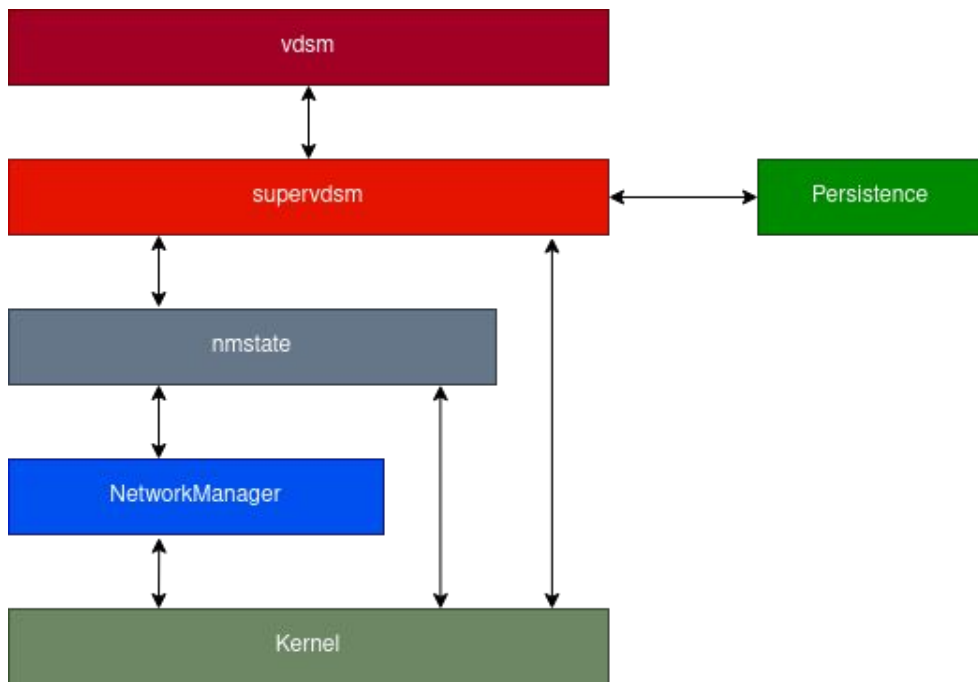
- Declarative network manager API for hosts
- Library with command line tool
- Network state is described in form of schema
- Multiprovider support
 - Currently only supported provide is NetworkManager
- Transaction and verification support

Architecture of oVirt



- Engine
 - Setup Networks through Webadmin or REST API
 - Save the configuration into database
 - Send configuration to the host

Architecture of vdsmd



- **Vdsm**
 - Forward Setup Networks to supervdsm
- **Supervdsm**
 - Translate the request into desired state
 - Save the persisted config on the host
- **Nmstate**
 - Apply the desired state via NetworkManager

Troubleshooting

```
root:api.py:220 Setting up network according to configuration: networks: {'test-network': {'nic':  
'dummy_tLKdn', 'switch': 'legacy'}}, bondings:{}, options: {'connectivityCheck': False}
```

```
root:configurator.py:190 Processing setup through nmstate
```

```
root:configurator.py:192 Desired state: {'interfaces': [{'name': 'dummy_tLKdn', 'state': 'up', 'mtu':  
1500, 'ipv4': {'enabled': False}, 'ipv6': {'enabled': False}}, {'name': 'test-network', 'type':  
'linux-bridge', 'state': 'up', 'mtu': 1500, 'bridge': {'port': [{'name': 'dummy_tLKdn'}]}, 'options': {'stp':  
{'enabled': False}}}, {'name': 'test-network', 'type': 'linux-bridge', 'state': 'up', 'mtu': 1500,  
'bridge': {'port': [{'name': 'dummy_tLKdn'}]}, 'options': {'stp': {'enabled': False}}}], 'ipv4': {'enabled': False}, 'ipv6': {'enabled': False}}}
```

```
root:nmclient.py:139 NM action queue exhausted, quitting mainloop
```

Troubleshooting

```
configurator::192::root:_setup_nmstate) Desired state:
{'interfaces': [{'name': 'bond0', 'type': 'bond', 'state': 'up',
'link-aggregation': {'slaves': ['ens4f1', 'ens4f2', 'ens4f3'],
options':{'miimon': '100', 'xmit_hash_policy': '2'},
'mode':'802.3ad'}, 'ipv4':{'enabled': False},'ipv6': {'enabled':
False}},{'name': 'ovirtmgmt'}]}
```

```
nmclient::201::root:(quit) NM main-loop aborted: Connection
update failed: error=nm-connection-error-quark:
ipv6.ra-timeout: unknown property (7), dev=ovirtmgmt/<enum
NM_DEVICE_STATE_ACTIVATED of type NM.DeviceState>
```

- Detailed setup networks log is in `/var/log/vdsm/supervdsm.log`
- Look for **NmstateVerificationError** or **NM main-loop aborted**
- Log contains the difference in case of verification error
- Main-loop aborted has hint above the abort between desired state or directly in the message.

Troubleshooting with nmstate

Before reproduction:

1. Set `RateLimitBurst=0` and `RateLimitIntervalSec=0` in `/etc/systemd/journald.conf`
2. Set `level=TRACE` and `domains=ALL` ([logging] section) in `/etc/NetworkManager/NetworkManager.conf`
3. `systemctl restart systemd-journald NetworkManager`

Reproduction:

1. Get problematic desired state from `supervdsm.log`
2. Copy it into some file e.g. `state.json`
3. Replace invalid JSON identifiers/characters e.g. `sed -i -e 's/\"/"/g' -e 's/False/false/g' -e 's/True/true/g' state.json`
4. Run it through `nmstatectl` e.g. `nmstatectl set state.json`

Implications

- Switch from network-scripts to nmstate should be transparent for users and only relevant for troubleshooting
- Import of Cockpit and Anaconda prepared network config by RHV is more fluent
- After the host is added to the oVirt engine, only oVirt engine must be used to modify the network of the host

Feature overview

- Through nmstate

- Networks (Bridged / Bridgeless)
- Bonds
- VLAN
- IP configuration (Static / Dynamic)
- MTU
- Default route
- DNS
- Custom Bond options

- Outside of nmstate

- Source routing / Policy based routing (Dynamic / Static)
- Custom Bridge options
- SR-IOV
- OvS
- QoS
- Refresh Capabilities

oVirt

Thank you!

<https://ovirt.org/>

users@ovirt.org

 @ovirt