목차

ovs for legacy network-script

Configure start up scripts for OVS on Centos and Red Hat Posted On September 13, 2016 The RPM packages for Open vSwitch provide some integration with RedHat’s network scripts. Using this integration is optional.

To use the integration for a Open vSwitch bridge or interface named <name>, create or edit /etc/sysconfig/network-scripts/ifcfg-<name>. This is a shell script that consists of a series of VARIABLE=VALUE assignments. The following OVS-specific variable names are supported:

1. DEVICETYPE: Always set to “ovs”.
2. TYPE: If this is “OVSBridge”, then this file represents an OVS bridge named <name>. Otherwise, it represents a port on an OVS bridge and TYPE must have one of the following values:
    - OVSPort, if <name> is a physical port (e.g. eth0) or virtual port (e.g. vif1.0).
    - OVSIntPort, if <name> is an internal port (e.g. a tagged VLAN).
    - OVSBond, if <name> is an OVS bond.
    - OVSTunnel, if <name> is an OVS tunnel.
    - OVSPatchPort, if <name> is a patch port
3. OVS_BRIDGE: If TYPE is anything other than “OVSBridge”, set to the name of the OVS bridge to which the port should be attached.
4. OVS_OPTIONS: Optionally, extra options to set in the “Port” table when adding the port to the bridge, as a sequence of column[:key]=value options. For example, “tag=100” to make the port an access port for VLAN 100. See the documentation of “add-port” in ovs-vsctl(8) for syntax and the section on the Port table in ovs-vswitchd.conf.db(5) for available options.
5. OVS_EXTRA: Optionally, additional ovs-vsctl commands, separated by “–” (double dash).
6. BOND_IFACES: For “OVSBond” interfaces, a list of physical interfaces to bond together.
7. OVS_TUNNEL_TYPE: For “OVSTunnel” interfaces, the type of the tunnel.

For example, “gre”, “vxlan”, etc.

1. OVS_TUNNEL_OPTIONS: For “OVSTunnel” interfaces, this field should be used to specify the tunnel options like remote_ip, key, etc.
2. OVS_PATCH_PEER: For “OVSPatchPort” devices, this field specifies the patch's peer on the other bridge.

주의사항

- “ifdown” on a bridge will not bring individual ports on the bridge down. “ifup” on a bridge will not add ports to the bridge. This behavior should be compatible with standard bridges (with TYPE=Bridge).
- If 'ifup' on an interface is called multiple times, one can see “RTNETLINK answers: File exists” printed on the console. This comes from ifup-eth trying to add zeroconf route multiple times and is harmless.

사용예제

Standalone bridge:

```
### ifcfg-ovsbridge0 ###
DEVICE=ovsbridge0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSBridge
BOOTPROTO=static
IPADDR=A.B.C.D
NETMASK=X.Y.Z.0
HOTPLUG=no
STP=on
```

Enable DHCP on the bridge:

- Needs OVSBOOTPROTO instead of BOOTPROTO.
- All the interfaces that can reach the DHCP server as a space separated list in OVSDHCPINTERFACES.

```
DEVICE=ovsbridge0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSBridge
OVSBOOTPROTO="dhcp"
OVSDHCPINTERFACES="eth0"
HOTPLUG=no
```

Adding Internal Port to ovsbridge0:

```
### ifcfg-intbr0 ###
DEVICE=intbr0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSIntPort
OVS_BRIDGE=ovsbridge0
HOTPLUG=no
```

Internal Port with fixed IP address:

```
DEVICE=intbr0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSIntPort
OVS_BRIDGE=ovsbridge0
BOOTPROTO=static
```

```
IPADDR=A.B.C.D
NETMASK=X.Y.Z.0
HOTPLUG=no
```

Internal Port with DHCP:

- Needs OVSBOOTPROTO or BOOTPROTO.
- All the interfaces that can reach the DHCP server as a space separated list in OVSDHCPINTERFACES.

```

DEVICE=intbr0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSIntPort
OVS_BRIDGE=ovsbridge0
OVSBOOTPROTO="dhcp"
OVSDHCPINTERFACES="eth0"
HOTPLUG=no
```

Adding physical eth0 to ovsbridge0 described above:

```

### ifcfg-eth0 ###
DEVICE=eth0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSPort
OVS_BRIDGE=ovsbridge0
BOOTPROTO=none
HOTPLUG=no
```

Tagged VLAN interface on top of ovsbridge0:

```

### ifcfg-vlan100 ###
DEVICE=vlan100
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSIntPort
BOOTPROTO=static
IPADDR=A.B.C.D
NETMASK=X.Y.Z.0
OVS_BRIDGE=ovsbridge0
OVS_OPTIONS="tag=100"
OVS_EXTRA="set Interface $DEVICE external-ids:iface-id=$(hostname -s)-
$DEVICE-vif"
HOTPLUG=no
```

Bonding:

```
### ifcfg-bond0 ###
DEVICE=bond0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSBond
OVS_BRIDGE=ovsbridge0
BOOTPROTO=none
BOND_IFACES="gige-1b-0 gige-1b-1 gige-21-0 gige-21-1"
OVS_OPTIONS="bond_mode=balance-tcp lacp=active"
HOTPLUG=no

### ifcfg-gige-* ###
DEVICE=gige-*
ONBOOT=yes
HOTPLUG=no
```

An Open vSwitch Tunnel:

```
### ifcfg-gre0 ###
DEVICE=ovs-gre0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSTunnel
OVS_BRIDGE=ovsbridge0
OVS_TUNNEL_TYPE=gre
OVS_TUNNEL_OPTIONS="options:remote_ip=A.B.C.D"
```

Patch Ports:

```
### ifcfg-patch-ovs-0 ###
DEVICE=patch-ovs-0
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSPatchPort
OVS_BRIDGE=ovsbridge0
OVS_PATCH_PEER=patch-ovs-1

### ifcfg-patch-ovs-1 ###
DEVICE=patch-ovs-1
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSPatchPort
OVS_BRIDGE=ovsbridge1
OVS_PATCH_PEER=patch-ovs-0
```

fake bridge configuration

```
DEVICE=vlan65
DEVICETYPE=ovs
TYPE=OVSBridge
ONBOOT=yes
BOOTPROTO=static
STP=off
NM_CONTROLLED=no
HOTPLUG=no
OVS_EXTRA="br-set-external-id $DEVICE bridge-id $DEVICE"
OVS_OPTIONS="br0 65"
```

Variation of fake bridge

```
### ifcfg-vlan100 ###
DEVICE=vlan12
ONBOOT=yes
DEVICETYPE=ovs
TYPE=OVSIntPort
BOOTPROTO=static
IPADDR=A.B.C.D
NETMASK=X.Y.Z.0
OVS_BRIDGE=ovsbr
OVS_OPTIONS="tag=12"
OVS_EXTRA="set Interface $DEVICE external-ids:iface-id=$(hostname -s)-
$DEVICE-vif"
HOTPLUG=no
```

참조링크

- https://github.com/openvswitch/ovs/blob/master/rhel/README.RHEL.rst