Establishing a Tunnel Using a Ramp Node

In some cases, the previous solution is not possible. For example, an F5 BIG-IP® host cannot run an OpenShift node instance or the OpenShift SDN because F5® uses a custom, incompatible Linux kernel and distribution.

Instead, to enable F5 BIG-IP® to reach pods, you can choose an existing node within the cluster network as a ramp node and establish a tunnel between the F5 BIG-IP® host and the designated ramp node. Because it is otherwise an ordinary OpenShift node, the ramp node has the necessary configuration to route traffic to any pod on any node in the cluster network. The ramp node thus assumes the role of a gateway through which the F5 BIG-IP® host has access to the entire cluster network.

Following is an example of establishing an ipip tunnel between an F5 BIG-IP® host and a designated ramp node.

**On the F5 BIG-IP® host:**

1. **Set the following variables:**

# F5\_IP=10.3.89.66 (The F5\_IP and RAMP\_IP variables refer to the F5 BIG-IP® host’s and the ramp node’s IP addresses, respectively, on a shared, internal network.)

# RAMP\_IP=10.3.89.89 (The F5\_IP and RAMP\_IP variables refer to the F5 BIG-IP® host’s and the ramp node’s IP addresses, respectively, on a shared, internal network.)

# TUNNEL\_IP1=10.3.91.216 (An arbitrary, non-conflicting IP address for the F5® host’s end of the ipip tunnel.)

# CLUSTER\_NETWORK=10.1.0.0/16 (The overlay network CIDR that the Openshift SDN uses to assign addresses to pods.)

1. **Delete any old route, self, tunnel and SNAT pool:**

# tmsh delete net route $CLUSTER\_NETWORK || true

# tmsh delete net self SDN || true

# tmsh delete net tunnels tunnel SDN || true

# tmsh delete ltm snatpool SDN\_snatpool || true

1. **Create the new tunnel, self, route and SNAT pool and use the SNAT pool in the virtual servers:**

# tmsh create net tunnels tunnel SDN \

\{ description "OpenShift SDN" local-address \

$F5\_IP profile ipip remote-address $RAMP\_IP \}

# tmsh create net self SDN \{ address \

${TUNNEL\_IP1}/24 allow-service all vlan SDN \}

# tmsh create net route $CLUSTER\_NETWORK interface SDN

# tmsh create ltm snatpool SDN\_snatpool members add { $TUNNEL\_IP1 }

# tmsh modify ltm virtual ose-vserver source-address-translation { type snat pool SDN\_snatpool }

# tmsh modify ltm virtual https-ose-vserver source-address-translation { type snat pool SDN\_snatpool }

**On the ramp node:**

1. **Set the following variables:**

# F5\_IP=10.3.89.66

# TUNNEL\_IP1=10.3.91.216

# TUNNEL\_IP2=10.3.91.217 (A second, arbitrary IP address for the ramp node’s end of the ipip tunnel.)

1. **Delete any old tunnel:**

# ip tunnel del tun1 || true

1. **Create the ipip tunnel on the ramp node, using a suitable L2-connected interface (e.g., eth0):**

# ip tunnel add tun1 mode ipip \

remote $F5\_IP dev eth0

# ip addr add $TUNNEL\_IP2 dev tun1

# ip link set tun1 up

# ip route add $TUNNEL\_IP1 dev tun1

# ping -c 5 $TUNNEL\_IP1

1. **SNAT the tunnel IP with an unused IP from the SDN subnet:**

# source /etc/openshift-sdn/config.env

# subaddr=$(echo $OPENSHIFT\_SDN\_TAP1\_ADDR | cut -d "." -f 1,2,3)

# export RAMP\_SDN\_IP=${subaddr}.254

1. **Assign this RAMP\_SDN\_IP as an additional address to tun0 (the local SDN’s gateway):**

# ip addr add ${RAMP\_SDN\_IP} dev tun0

1. **Modify the OVS rules for SNAT:**

# ovs-ofctl -O OpenFlow13 add-flow br0 \

"cookie=0x999,ip,nw\_src=${TUNNEL\_IP1},actions=mod\_nw\_src:${RAMP\_SDN\_IP},resubmit(,0)"

# ovs-ofctl -O OpenFlow13 add-flow br0 \

"cookie=0x999,ip,nw\_dst=${RAMP\_SDN\_IP},actions=mod\_nw\_dst:${TUNNEL\_IP1},resubmit(,0)"

# ovs-ofctl -O OpenFlow13 add-flow br0 \

"cookie=0x999, table=0, arp, arp\_tpa=${RAMP\_SDN\_IP}, actions=output:2"

1. **Mark the ramp node as an unschedulable node so that no pods end up on the ramp node itself:**

$ oadm manage-node <ramp\_node\_hostname> --schedulable=false

Configuring a Highly-Available Ramp Node

You can use OpenShift’s ipfailover feature, which uses keepalived internally, to make the ramp node highly available from F5 BIG-IP®'s point of view. To do so, first bring up two nodes, for example called ramp-node-1 and ramp-node-2, on the same L2 subnet.

Then, choose some unassigned IP address from within the same subnet to use for your virtual IP, or VIP. This will be set as the RAMP\_IP variable with which you will configure your tunnel on F5 BIG-IP®.

For example, suppose you are using the 10.20.30.0/24 subnet for your ramp nodes, and you have assigned 10.20.30.2 to ramp-node-1 and 10.20.30.3 to ramp-node-2. For your VIP, choose some unassigned address from the same 10.20.30.0/24 subnet, for example 10.20.30.4. Then, to configure ipfailover, mark both nodes with a label, such as f5rampnode:

$ oc label node ramp-node-1 f5rampnode=true

$ oc label node ramp-node-2 f5rampnode=true

Similar to instructions from the ipfailover documentation, you must now create a service account and add it to the privileged SCC. First, create the f5ipfailover service account:

$ echo '

{ "kind": "ServiceAccount",

"apiVersion": "v1",

"metadata": { "name": "f5ipfailover" }

}

' | oc create -f -

Next, you can manually edit the privileged SCC and add the f5ipfailover service account, or you can script editing the privileged SCC if you have jq installed. To manually edit the privileged SCC, run:

$ oc edit scc privileged

Then add the f5ipfailover service account in form system:serviceaccount:<project>:<name> to the users section:

...

users:

- system:serviceaccount:openshift-infra:build-controller

- system:serviceaccount:default:router

- system:serviceaccount:default:f5ipfailover

Alternatively, to script editing privileged SCC if you have jq installed, run:

$ oc get scc privileged -o json |

jq '.users |= .+ ["system:serviceaccount:default:f5ipfailover"]' |

oc replace scc -f -

Finally, configure ipfailover using your chosen VIP (the RAMP\_IP variable) and the f5ipfailover service account, assigning the VIP to your two nodes using the f5rampnode label you set earlier:

# RAMP\_IP=10.20.30.4

# IFNAME=eth0 (The interface where RAMP\_IP should be configured.)

# oadm ipfailover <name-tag> \

--virtual-ips=$RAMP\_IP \

--interface=$IFNAME \

--watch-port=0 \

--replicas=2 \

--service-account=f5ipfailover \

--selector='f5rampnode=true'

With the above setup, the VIP (the RAMP\_IP variable) is automatically re-assigned when the ramp node host that currently has it assigned fails.