

[Generally Useful Info](#) >

HOWTO: Shared Interface for Management and Data Plane

In certain deployments, it may be useful to share a physical device interface for 128T data plane, and router management. This HOWTO describes a setup for sharing a single device interface for management and routing, using 128T services and KNI.

Setup

This setup uses a typical branch office deployment as an example of how to configure a 128T Router to use a single interface for routing and management. The system hosting 128T has 2 device interfaces:

- **eth0** connected to the LAN for system management and LAN gateway
- **eth1** connected to the WAN/ISP

For this example our LAN network is **192.168.1.0/24**, and the IP address for gateway and router management is **192.168.1.1**.

Configuration

In order to allow for access to the system at any time, we must account for two different networking states the system might be in:

- 128T Up/running
- 128T Down/stopped

128T Down

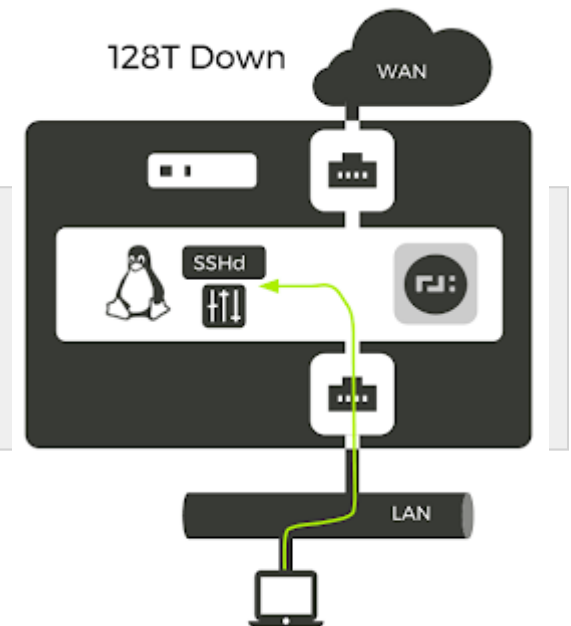
For the case that 128T is down, Linux will have control the **eth0** interface. The following network script assigns the appropriate address and network information to the interface:

```
[t128@host ~]$ sudo cat /etc/sysconfig/network-scripts/ifcfg-eth0
BOOTPROTO=static
IPV6INIT=no
NAME=eth0
DEVICE=eth0
ONBOOT=yes
ZONE=trusted
IPADDR=192.168.1.1
NETMASK=255.255.255.0
```

With this configuration, administrators should be able to access the Linux host with SSH from the LAN network, by connecting to the **192.168.1.1** address.

128T Up

When 128T is up and running, **eth0** will no longer be visible to the Linux host. Therefore we must use a Kernel Network Interface (KNI) between the 128T data plane and the Linux Kernel for access. The following configuration sets up the gateway interface, and uses a link local address and a /31



segment for the KNI:

```
device-interface 1
  id 1
  description "NIC connect to LAN"
  type ethernet
  pci-address 0000:0b:00.3

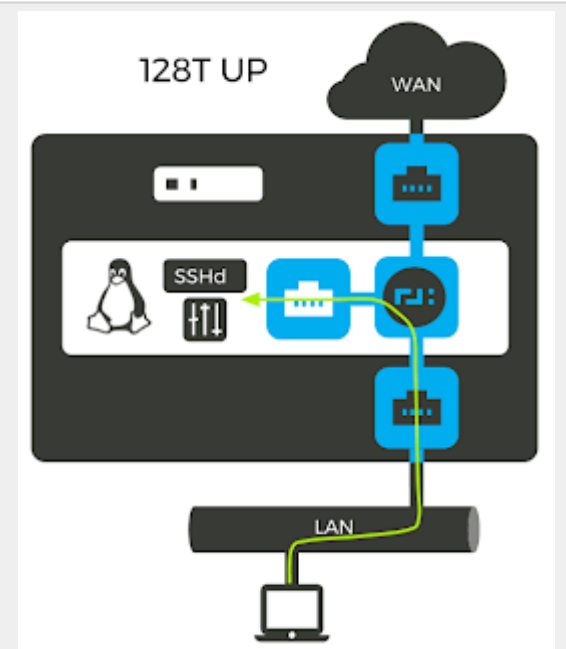
  network-interface lan
    name lan
    description "LAN gateway"
    tenant my-tenant

    address 192.168.1.1
    ip-address 192.168.1.1
    prefix-length 24
  exit
exit
exit

device-interface 128
  id 128
  description "KNI for management"
  type kni

  network-interface mgmt-kni
    name mgmt-kni
    source-nat true

    address 169.254.128.0
    ip-address 169.254.128.0
    prefix-length 31
  exit
exit
exit
```



***Note:** See that the configuration is set with `source-nat true`. This is important because while this configuration will instantiate a KNI interface in Linux, it will not automatically set up any routes in Linux to use the KNI interface. This means that when packets come in via the KNI from the LAN (`192.168.1.0/24`), Linux will not have a route to send responses back. By enabling source-nat on the KNI, packets arriving via the KNI will be sourced from `169.254.128.0` instead. Linux can easily route responses to this address, since it is local connected.

Of course 128T will not pass any packets to this interface, without some kind of service and associated policy defined. The following configuration establishes a service to handle connections to the `192.168.1.1` address for SSH (tcp port 22) and HTTPS (tcp port 443):

```
service mgmt
  name mgmt
  description "Router managment"
  tenant my-tenant

  transport tcp
  protocol tcp

  port-range 22
  start-port 22
exit
```

```
        port-range 443
        start-port 443
    exit
exit
address 192.168.1.1/32
exit
```

***Note:** This configuration places the service in the "my-tenant" tenant, which is the same tenant set on the LAN network interface. This provides implicit access to the `mgmt` service from users in the LAN network.

Last but not least, we need a route to send connections to Linux through the KNI:

```
service-route sr-mgmt
name sr-mgmt
service-name mgmt
destination 169.254.128.1

next-hop my-node mgmt-kni 169.254.128.1
node-name my-node
interface mgmt-kni
gateway-ip 169.254.128.1
exit
exit
```

128T Router GUI Considerations

By default the 128T webserver will use the address set in its `global.init` to listen for connections. So that the webserver can be accessed from the LAN when 128T is running, it must also listen on the KNI interface. The following configuration will cause the webserver to listen on any interfaces available to Linux (including the KNI):

```
system
services
    webserver
        enabled true

        server my-node
            node-name my-node
            ip-address 0.0.0.0
        exit
    exit
exit
exit
```

Verifying Operation

To verify operation, you should be able to connect to Linux using SSH at `192.168.1.1` when 128T is up or down. When 128T is up, the router GUI should also be available at `192.168.1.1`.

Note that when stopping or starting, your SSH connections will hang, and must be re-established.

Also when 128T is running with this configuration, you can verify the KNI interface in Linux:

```
[t128@host ~]$ ip a
2: kni128: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 6e:3e:3c:3e:de:c1 brd ff:ff:ff:ff:ff:ff
    inet 169.254.128.1/31 scope global kni8
        valid_lft forever preferred_lft forever
    inet6 fe80::6c3e:3cff:fe3e:dec1/64 scope link
        valid_lft forever preferred_lft forever
```