Quick Guide: Using link.py for Veth Pair Management in Incus Containers

Your Name

November 13, 2024

Introduction

This document provides a quick guide on using `link.py` to create and manage virtual Ethernet (veth) pairs in Incus containers. Veth pairs are useful for creating network connections between different network namespaces, such as those used by containers.

Requirements

- **Python** (preferably version 3.x)

- **Root privileges**: Run `link.py` with `sudo` to ensure the necessary permissions.

- **Incus installed** and properly configured on your system.

- **Running containers**: Ensure that the Incus containers you want to connect are running.

Basic Usage

The general syntax to run `link.py` is as follows:

```
sudo python link.py \
    -ns1 <namespace1> -t1 <type1> \
    -ns2 <namespace2> -t2 <type2> \
    -n1 <veth1> -n2 <veth2> \
    [-b1 <bridge1>] [-b2 <bridge2>]
```

- `-ns1`, `-ns2`: Names of the network namespaces or containers.

- `-t1`, `-t2`: Container type (use `incus` for Incus containers).

- `-n1`, `-n2`: Names of the veth interfaces.

- `-b1`, `-b2` (optional): Attach veth to a specified bridge on either end.

Example Commands

Example 1: Connect Host and Incus Container

Connect the host network namespace to an Incus container's network namespace.

```
sudo python link.py \
    -ns1 my_incus_container -t1 incus \
    -n1 veth_container -n2 veth_host
```

Example 2: Connect Two Incus Containers

Create a veth pair connecting two Incus containers.

```
sudo python link.py \
    -ns1 incus_container1 -t1 incus \
    -ns2 incus_container2 -t2 incus \
    -n1 veth1 -n2 veth2
```

Example 3: Attach Host End to a Bridge

Create a veth pair between the host and an Incus container, attaching the host end to a bridge named `br0`.

```
sudo python link.py \
    -ns1 my_incus_container -t1 incus \
    -n1 veth_container -n2 veth_host \
    -b2 br0
```

Notes

- If '1' is specified for `-ns1` or `-ns2`, it defaults to the host namespace.

- Ensure bridges exist before attempting to attach veth pairs to them.

- The script automatically brings up interfaces and bridges after creation.

Testing Connectivity

Assign IP addresses to each end of the veth pair for testing connectivity.
Example commands:

```
# On the host
sudo ip addr add 192.168.10.1/24 dev veth_host

# Inside the Incus container
sudo incus exec my_incus_container -- ip addr add
    192.168.10.2/24 dev veth_container

# Test with ping
ping 192.168.10.2
```

Troubleshooting

- **Permission errors**: Ensure you're running the script with `sudo`.

- **Interface not found**: Verify that the interface names are unique and
  do not conflict with existing interfaces.

- **Container not found**: Check that the Incus container names are
  correct and that they are running.