Network Virtualization Tools in Linux

PRESENTED BY:
QUAMAR NIYAZ & AHMAD JAVAID
Contents

- Introduction
- Types of Virtualization
- Network Virtualization
- OS Virtualization
- OS-Level Virtualization
- Some Virtualization Tools + Features
- Some Virtualization Scenarios
Introduction

- Virtualization:
  - An illusion of resource ownership

- Exists since 1960s
  - First used in Mainframes
  - OS level illusion to the processes about CPU, memory, etc. ownership

- Meaning **broadened** in last few decades
Types of Virtualization

- Various kinds of virtualizations:
  - OS Virtualization
  - Network Virtualization
  - Application Level Virtualization
  - Hardware Virtualization
  - ... and many more

- Our Focus: Network Virtualization
  - Will also use OS Virtualization
Network Virtualization

- Abstraction of physical network
  - Co-existence of several virtual networks on a single physical network

- Involves platform/resource virtualization
  - Resources required to interconnect
  - We use OS Virtualization
OS Virtualization

- Sometimes categorized as
  - Platform Virtualization
  - Computer Hardware Virtualization

- Several Categories defined, major ones listed here
  - Full-Virtualization
    - uses hypervisors or VMMs
    - Guest OS is unaware of the virtualized environment
  - Para-Virtualization
    - Modified guest kernel
    - Hypervisor provides hypercall interfaces
  - OS-level Virtualization (Mostly Linux based)
OS Virtualization (contd.)

- Tools for Full and Para virtualization
  - Available for Linux + Windows
  - VMware
  - Hyper-V
  - Xen

- Visual Example
OS-level Virtualization

- The guest OS shares the host kernel
- Does not use any hypervisor
  - Executes directly on the host OS
- Performs better compared to other technique
- Mostly Linux based
- Definitely comes with limitations ... !!
OS-level Virtualization

OS-level Virtualization tools

- Linux Container (LXC)
  - Two Linux kernel features needed
    - **Namespace**: gives illusion of isolated instance to processes within itself
    - **Cgroups** (control groups): used to limit, account, and isolate resource usage of process groups

- Linux-V server
- OpenVZ
Network Namespace

- Six kinds of namespace in Linux
  - Mount, UTS, IPC, PID, Network, Username

- Network namespace
  - provides isolation of system resources associated with networking
  - different and separate instances of network interfaces and routing table.
Network namespace (contd.)

- Creating and listing network namespace
  - `ip netns add vnet`
  - `ip netns list`

- Assigning Interfaces network namespace
  - `ip link add veth0 type veth peer name veth1`
  - `ip link set veth1 netns vnet`

- Configuring interfaces in network namespace
  - `ip netns exec vnet ifconfig veth1 10.1.1.11/24 up`

- Access a network namespace
  - `ip netns exec vnet bash`
Linux Containers (LXC)

- Light-weight virtualization
  - in between chroot and a full fledged virtual machine.

- Goal
  - Create a close to standard Linux environment

- Based on
  - namespace isolation and cgroup

- Allows to run a complete Linux instance
Linux Containers (contd.)

- LXC can be installed from Ubuntu repository
  - `sudo apt-get install lxc`

- Creating an LXC
  - From template: `sudo lxc-create -n ub1 -t Ubuntu`
  - Clone: `lxc-clone -n ub2 -o ub1`

- Start/Shut-down an LXC
  - `lxc-start -n ub1`
  - `lxc-shutdown -n ub1`
Linux Containers (contd.)

- LXC Configuration available in its directory
  - /var/lib/lxc/ub1/config

- Edit the config file for network settings
  - lxc.network.link = br0
  - lxc.network.type = veth
  - lxc.network.flags = up
  - lxc.network.veth.pair = c1eth1
  - lxc.network.ipv4 = 10.10.1.11/24
Open vSwitch (OVS)

- Open source software switch
  - Implements basic Ethernet switch
  - Capable of being a production quality, multi-layer virtual switch at its extreme

- Can operate both as
  - software switch
  - control stack for programmable hardware switches

- Implements open flow protocol for SDNs
Features of OVS

- **Security**
  - VLAN isolation, traffic filtering

- **Monitoring**
  - NetFlow, sFlow, SPAN, RSPAN

- **QoS**
  - Traffic queuing, traffic shaping

- **Automated Control**
  - Open flow, ovsdb mgmt. protocol
OVS utilities

- **Open vswitchd management utility:** `ovs-vsctl`
  - `list-br`
  - `add-br br0`
  - `add-port br0 eth0`
  - `show br0`

- **Open-flow enabled switch management utility:** `ovs-ofctl`
Network Virtualization Scenarios

- Full-virtualization based VMs
- Networking features
  - Bridged mode, NAT, Host only
- VMs attached with Open vSwitch ports
- Network namespace and its integration with OVS
- LXC and its integration with OVS
Hands-On Scenario

Host Machine 1

- Virtual Host 1
  10.10.1.11
  eth0
  veth1
  Br0(10.10.1.1)

Host Machine 2

- Virtual Host 1
  10.10.1.21
  eth0
  veth1
  Br0(10.10.1.2)

- Virtual Host 2
  10.10.1.12
  eth0
  veth2

- Virtual Host 2
  10.10.1.22
  eth0
  veth2
Thank You!