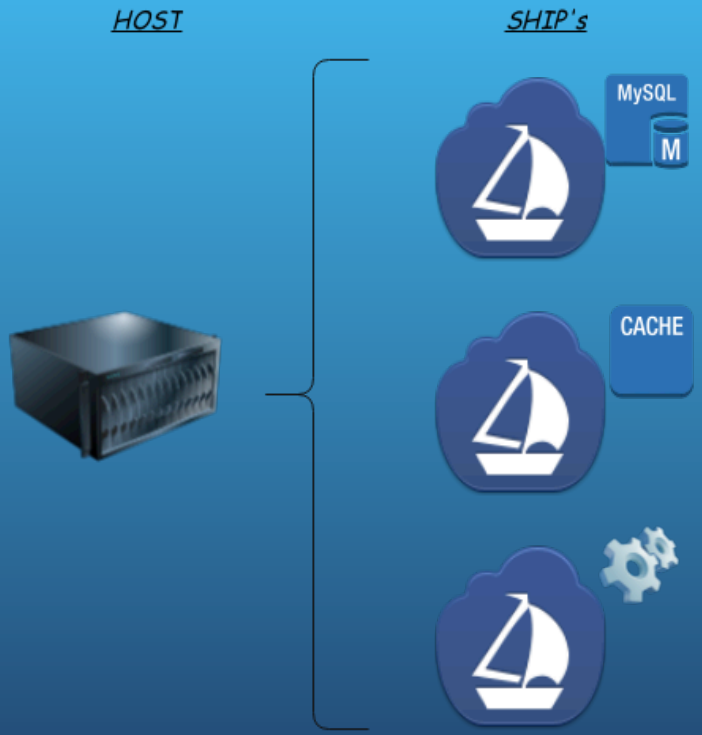
SHIP

The true about me

Alberto Ferreira
bsurfer@gmail.com

https://github.com/bsurfer/ship

Purpose



- Have various services in same host
- Shared resources
- Isolated services
- Isolated IP Address

What is?

- Is nothing more than a, normal, chroot where you can run distributed applications and services
- Enable services to be quickly install and configured

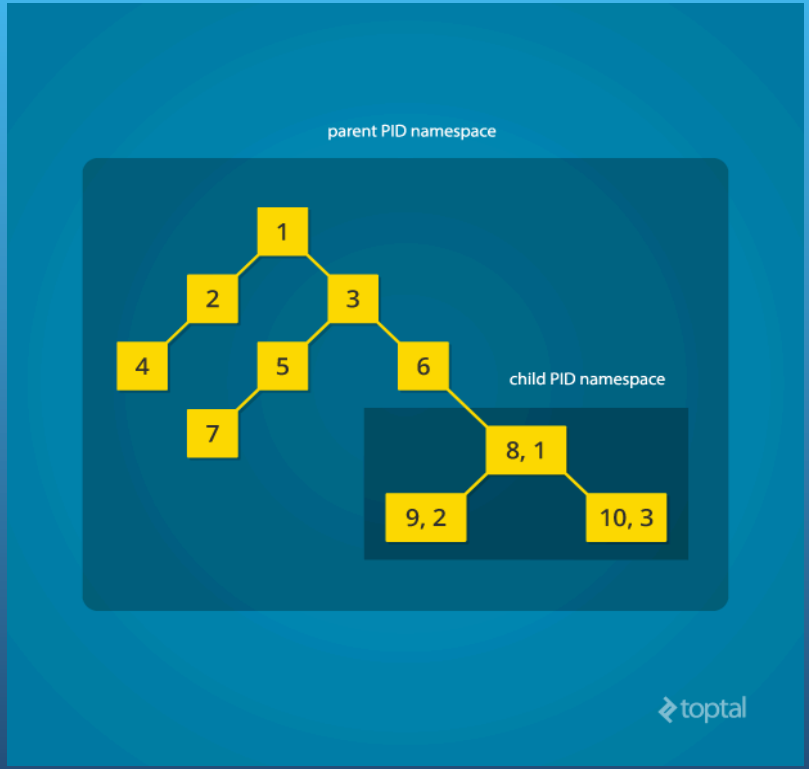
Features

- Run like a normal SO
- All services was isolated
- /proc and /sys are host independent
- Is possible have one or more network interfaces
- CLI to start/stop/connect in to chip
- Portable
- Run different versions of debian OS

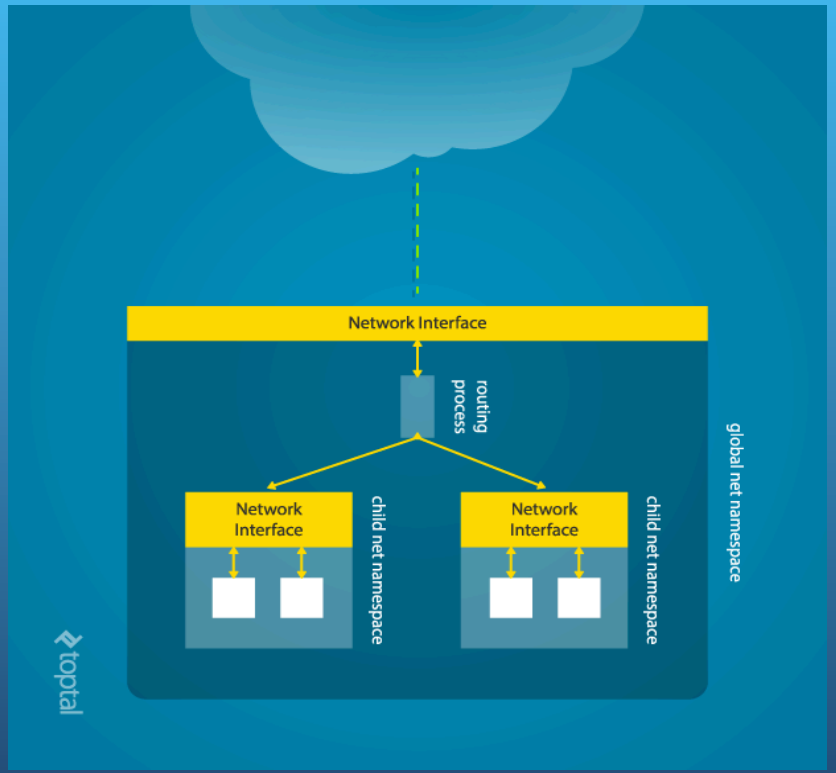
Tecnologies

- unshare
- namespaces
- nsenter
- openvswitch

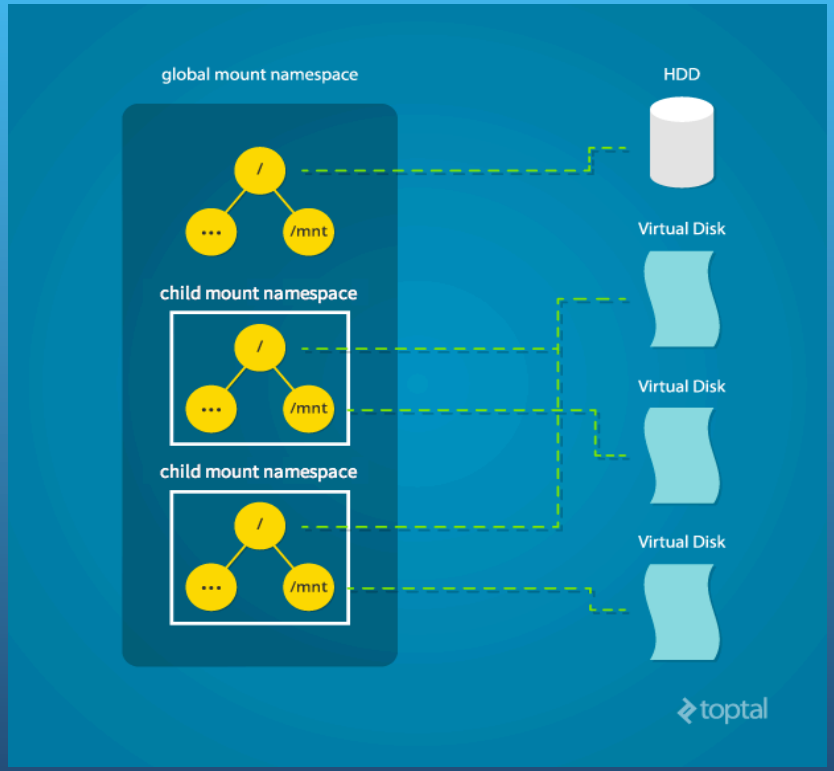
Process Namespace



Network Namespace



Mount Namespace



Challenges

- unshare, in Debian, the option -p and -f is missing by default
- Start a ship like a normal startup
- Ship’s can’t communicate with the host, via ssh

Code tree

```
ship
|-- bin
|   |-- nsenter
|   `-- unshare
|-- config
|   `-- ship.cfg
|-- config_files
|   |-- interfaces
|   |-- interfaces_ovs
|   `-- start_chroot
`-- scripts
    |-- shipmn
    `-- shipmns
```

Some code

Config File:

```
f = file('/etc/ship/config/ship.cfg')
cfg = Config(f)
farm_folder=cfg.farm_folder
config_folder=cfg.config_folder
pid_folder=cfg.pid_folder
```

Some code (Cont)

- Check if IP is valid and if is being used:

```
def valid_ip(ip):
      try:
            socket.inet_aton(ip)
            return True
      except socket.error:
            return False


def ping_ip(ip):
     try:
               sh.ping(ip, "-c 1", _out="/dev/null")
               return True
     except sh.ErrorReturnCode_1:
               return False
```

Some code (Cont)

- Create network interface:

```
def create_network(chroot):
        print 'Create Network'
        os.system('ip link delete %s'  %(chroot) )
        os.system('ip link add %s type veth peer name veth0' %chroot)
        os.system('ovs-vsctl del-port ovsbr %s; ovs-vsctl add-port ovsbr %s'  %(chroot,chroot) )
        os.system('ifconfig %s promisc' %chroot)


def create_network_stage2(pid,interface):
        print 'Put interfaces %s UP!!! PID: %s' %(interface,pid)
        os.system('ip link set netns %s veth0' %pid)
        os.system('ifconfig %s promisc'  %interface)
        os.system('ifconfig %s up'  %interface)
```

Some code (Cont)

- Start Ship

```
start_network(value)
if get_interfaces(value) is '2':
            create_network_wan(value)
father_hostname(value)
cmd=('setsid %sbin/unshare -m -p -u -n -f chroot %s%s /etc/init.d/start_chroot ') %
     (config_folder,farm_folder,value)
print 'Launch Chroot'
p_pid = subprocess.Popen(['/bin/bash', '-c', cmd]).pid
time.sleep(1)
pid = get_start_pid(p_pid)
create_pid(pid,value)
create_network_stage2(pid,value)
if get_interfaces(value) is '2':
     create_network_stage2_wan(pid,value)
```

Some code (Cont)

- Stop Ship:

```
if status_chroot(value) is True:
                pid=int(get_pid(value))
                print 'Shutdown: %s Pid: %s' %(value,pid)
                os.kill(pid, signal.SIGTERM)
                delete_pid(value)
```

- Go inside the Ship:

```
if get_pid(name) is not False:
        cmd=('%sbin/nsenter -t %s  -m -u -p -n chroot %s%s ') %
    (config_folder,get_pid(name),farm_folder,name)
        os.system(cmd)
```

Usage

Usage: shipmn [options]

Options:

create	create new ship using debootstrap
status	ship's status
start all	start all ship's
start ship_name	start ship
stop all	stop all ship's
stop ship_name	stop ship
con ship_name	enter in ship

Future

- Web Interface:
  - Ship’s status
  - Statistics
  - Running process
  - Create/delete/clone Ship’s
- Cgroups integration
- Clone ship’s