Towards a Robust FreeBSD-Based Cloud: Porting OpenStack Components

Chih-Hsin Chang @ AsiaBSDCon 2024
Outlines

- Introduction
- Background
- Current Status
- Challenges
- Roadmap
- Conclusion
Outlines

- Introduction
- Background
- Current Status
- Challenges
- Roadmap
- Conclusion
Who Am I

- Chih-Hsin (Zespre) Chang
- Software developer @ SUSE
- Harvester HCI open source project
Project Origin

- CHERI (Capability Hardware Enhanced RISC Instructions)
  - Managing a set of Morello evaluation boards with OpenStack Ironic

- The OpenStack on FreeBSD Project
  - Started in Jan. 2022
  - Chih-Hsin Chang & Li-Wen Hsu (lwhsu)
  - Initially targeting OpenStack Ironic
  - Pivot to VM-first
Outlines

- Introduction
- Background
  - Keystone
  - Glance & Placement
  - Neutron
  - Nova
- Current Status
- Challenges
- Roadmap
- Conclusion
Open What?

- A cloud infrastructure for virtual machines, bare metal, and containers
- Consist of a stack of open-source software components to provide services
  - Compute
  - Networking
  - Storage
  - Orchestration
  - Application lifecycle
  - Telemetry
  - ...
- Latest release: 2023.02 Bobcat
Keystone (Identity Service and Service Catalog)

- API client authn and authz
- Support LDAP server as backend
- Service discovery
Glance (Image Service) & Placement (Inventory Service)

- Serve VM images and their metadata
- Track cloud resource inventory and usage
- Help other services, e.g. Nova, do the decision about resource allocation
Neutron (Networking Service)

- **API server**
  - Accept HTTP-based requests from other components

- **Various agents**
  - L2: L2 network connectivity to OpenStack resources
  - L3: virtual routers and floating IPs
  - DHCP: IP address issuance
  - Metadata: cloud-init metadata and user data

- **ML2 (Modular Layer 2) plug-ins**
  - Type drivers: flat, Geneve, GRE, VLAN, and VXLAN
  - Mechanism drivers: Open vSwitch, Linux bridge, OVN, SRIOV, MacVTap, and L2 population
Nova (Compute Service)

- **API server**
  - Accept HTTP-based requests from other components

- **Scheduler**
  - Collect resource usage from compute nodes
  - Decide what node to run the instance

- **Conductor**
  - Prepare instance information based on DB entries

- **Compute**
  - Manage instance lifecycle through hypervisor on each compute node
  - Hypervisor manager

- **Serial proxy**
  - Provide access to instance console over WebSocket
Ironic (Bare-metal Provisioning Service)

- Manages bare-metals in contrast to typical Nova usage
- Deployment models
  - Stand-alone mode
  - Keystone + Ironic
  - As a Nova virt driver
Outlines

- Introduction
- Background
- Project Status
  - Development Environment
  - OpenStack Xena Integration
  - Porting OpenStack Components
  - Demo
- Challenges
- Roadmap
- Conclusion
Development Environment

- In-house development environment
  - Processors: 2 x Intel® Xeon® E5-2680 v4
  - Motherboard: Supermicro® X10DRL-i
  - Memory: 64 GB RAM
  - Storage: 1 TB SSD

- Remote PoC site: openstack1
- Single-node, all-in-one cluster
So, what does it look like now?

- Install from source
- Each component runs in its own Python virtual environment

- Keystone
  - Source code: unmodified
- Glance
  - Source code: unmodified
- Placement
  - Source code: unmodified
- Neutron
  - Source code: patches
  - Configuration: flat network + Open vSwitch
- Nova
  - Source code: patches
  - Configuration: libvirt + bhyve

➢ Limitations
  - No tenant network isolation
  - Need external DHCP service
  - No floating IPs
(Live?) Demo

The demo video, just in case something bad happens
https://asciinema.org/a/647308
OpenStack Xena Integration

- The “OpenStack on FreeBSD” GitHub organization
  - [https://github.com/openstack-openstack](https://github.com/openstack-openstack)
  - Steps by step build and installation guide
    - [openstack-on-freebsd/docs](https://github.com/openstack-openstack/openstack-on-freebsd/docs)
  - Administration (issue management)
    - [openstack-on-freebsd/admin](https://github.com/openstack-openstack/openstack-on-freebsd/admin)
  - Ported source code
    - (forked) [openstack-on-freebsd/neutron](https://github.com/openstack-on-freebsd/neutron)
    - (forked) [openstack-on-freebsd/nova](https://github.com/openstack-on-freebsd/nova)
  - FreeBSD ports collection
    - [openstack-on-freebsd/openstack](https://github.com/openstack-on-freebsd/openstack)
  - Custom solutions
    - [openstack-on-freebsd/socat-manager](https://github.com/openstack-on-freebsd/socat-manager)
    - (forked) [openstack-on-freebsd/novaconsole](https://github.com/openstack-on-freebsd/novaconsole)
Outlines

- Introduction
- Background
- Current Status
- Challenges
  - Computing
  - Networking
  - Privilege Model
  - Miscellaneous
- Roadmap
- Conclusion
Challenge - Computing

- Nova abstracts the operations against the underlying hypervisors
- Nova virtualization driver
  - Well-defined interfaces
  - Per-compute node configuration
- Currently supported drivers
  - libvirt.LibvirtDriver
  - fake.FakeDriver
  - ironic.IronicDriver
  - vmwareapi.VMwareVCDriver
  - zvm.ZVMDriver
Using the libvirt Driver on FreeBSD

- **libvirt**
  - Only implement a limited set of functionalities for FreeBSD/bhyve

- **libvirt virt driver**
  - Some operations specific to bhyve not covered by libvirt
  - Require a new virtualization type - bhyve
Challenge - Networking

- The combination of ML2 drivers for FreeBSD
  - Type driver: flat
  - Mechanism driver: openvswitch

- L2 agent
  - No Linux bridge available
  - No iptables available

- L3 agent - virtual routers
  - No iptables

- DHCP agent
  - No Linux network namespace
  - No Linux veth pairs
Open vSwitch on FreeBSD

- Open vSwitch `datapath_type=netdev`, without DPDK
  - No openvswitch kernel module
  - The combination is considered experimental (not tested thoroughly)
  - Performance issue

- Todos
  - Enable DPDK
  - Develop native FreeBSD bridge agent
IP Address Mismatch

- VMs get IP addresses from the external DHCP server
- Flow rules enforced by the underlying Open vSwitch
  - Source IP address does not match the one Neutron allocated
- Result: packets originated from VMs get dropped
Challenge - Privilege Management

- Principle of least privilege
  - Running with reduced/no privilege
  - Escalating when absolutely required

- All operations will be translated into commands and run on the OS, eventually
  - `chown(8)`
  - `ip(8)`
  - `ovs-vsctl(8)`
The Evolution of Privilege Mechanism in OpenStack

- **sudo**
  - One-shot
  - All or nothing

- **oslo.rootwrap**
  - Allow advanced filters
  - Support one-shot or daemon mode
  - Performance penalty
  - Does not allow long-lived/streaming commands

- **oslo.privsep**
  - Leverage Linux capabilities
    - Drop root superpowers but only keep what is required
  - Two-process model (unprivileged and privileged)
    - Connected over a local communication channel
    - Share the same fate

```
$ sudo nova-rootwrap /etc/nova/rootwrap.conf command
```
What about FreeBSD?

- Linux capabilities is not available on FreeBSD
- Workaround
  - Fallback to rootwrap
- Formal solution
  - Leverage FreeBSD’s own privilege management mechanism
Misc - Exposing VM Serial Console

- Introducing socat-manager
  - Listening on Unix socket
  - Maintaining TCP port to \texttt{nmdm(4)} mappings
  - Managing \texttt{socat(1)} processes
    $$\$/usr/local/bin/socat \$
       \texttt{file:/dev/nmdm21B,ispeed=9600,ospeed=9600,raw,echo=0} \$
       \texttt{tcp-listen:10014,bind=0.0.0.0,reuseaddr,fork}$$

- The libvirt hook script
  - Taking the domain XML as the input
  - Calling socat-manager with parameters (port and nmdm device name) as the side effect

- Ugly, but it works
On Linux hosts

On FreeBSD hosts
Outlines

- Introduction
- Background
- Current Status
- Challenges
- Roadmap
- Conclusion
Roadmap

- Development of native drivers for Neutron and Nova
- Porting additional OpenStack components to FreeBSD
- Migration to new versions of OpenStack
- Creating corresponding FreeBSD ports
- Continuous engagement and knowledge sharing
- Performance and scalability improvements
Outlines

- Introduction
- Background
- Current Status
- Challenges
- Roadmap
- Conclusion
Conclusion

- Use cases are very limited
  - We dropped many things to make it viable
- There are many topics/issues need expertise
  - Exploring network implementation equivalents
  - Finding suitable privilege model
- Bringing Linux-first design to FreeBSD
- Follow the Windows path (?)
- Need to formalize the changes
Thank you!

starbops@hey.com