Cumulus Networks, Dell and Red Hat Simplify 300+ Node OpenStack Pod

1 — Layer 3 networking agility

- Routed IP networking removes layer 2 complexities and limitations

- Native controller-less VXLAN network virtualization

- Customizable number of links for desired level of overcommit, redundancy, and load balancing

2 — Scalability via NetDevOps
(standard Linux DevOps tools for networking)

- Controller-less SDN built on native Neutron

- Bridging the gap between Network Engineers and System Administrators

- Enables Infrastructure as Code principles

ANSIBLE

ECMP

ECMP   ECMP   ECMP   ECMP

L3
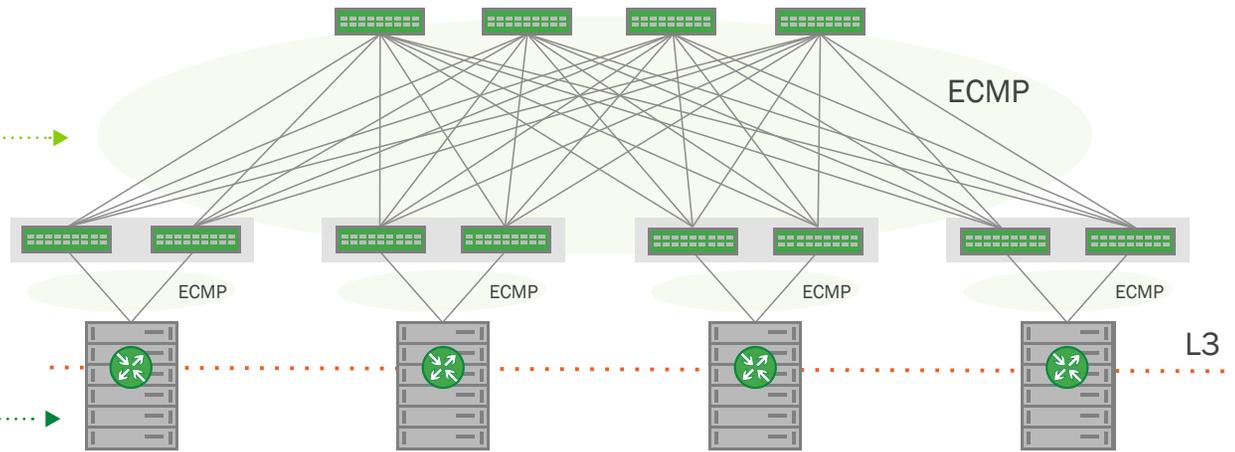
3 — Compute simplicity with Cumulus Linux Quagga package

- Easy to deploy and maintain compute nodes

- Simplified IP networking for System Administrators

- Enables plug-and-play deployment that is reliable and scalable

4 — Infrastructure scale deployed with open hardware

- Industry standard switch and server hardware running Linux all the way down

- Cumulus Networks Ready: Mitigate risk and vendor lock-in by leveraging 40+ platforms from 8 hardware vendors

- Choice at the ASIC level - Broadcom and Mellanox

Cumulus Networks, Dell and Red Hat Simplify 300+ Node OpenStack Pod

Overview

OpenStack is all about providing an Infrastructure as a Service solution platform complete with compute and network which is easy to deploy, configure and scale, while keeping it economical. Open compute standards are heavily utilized when building OpenStack clusters, but in most cases this doesn't extend to the network. Network Engineers and System Administrators are either purposely or unintentionally siloed due to a lack of common frameworks or legacy methodologies and mindsets. Hardware vendor lock-in and inflexible organizational structures prevent customers from taking full advantage of OpenStack's benefits.

The Cumulus® Linux® makes it easy to deploy, configure, and manage switches because the data and control plane now "speak the same language" with Linux all the way down.

In February 2016 Cumulus Networks, Dell, and Red Hat set out to apply network and server automation to a production-ready OpenStack pod while removing scalability and maintenance challenges for end users. The project demonstrated successful use of shared DevOps tools that tore down functional silos, fostered collaboration, and implemented infrastructure as code concepts.

In the end, the main objective was to build the largest production-ready OpenStack pod that was all Linux, all layer 3 networking, and all open while removing scalability and maintenance challenges.

Methodology Deployed without Constraints

Cumulus Networks, Dell and Red Hat built a 300+ node pod connected via six spine switches and eighteen leaf switches running layer 3 networking all the way down the rack.

Before starting the actual deployment, the team created a virtual prototype of the entire pod with Cumulus VX, ran all the simulations and then transferred the findings to the actual on-premise production environment.

The OpenStack pod with 100% Linux gave a common interface between computer and network resources. This enabled the team to leverage common DevOps tools like Ansible by Red Hat and Git across the pod, resulting in an easy, highly scalable and low touch solution.

Final Results

Having Linux all the way down the rack both physically and logically helped in the following ways:

- Created a common interface between Network Engineers and System Administrators, allowing existing automation tools to work across the entire pod.

- Made installation and provisioning easy, which was completed within six hours.

- Simplified IP networking and controller-less VXLAN-based SDN.

- Provided innovative ways to make networking available to compute nodes without complex SDN controllers. The Cumulus Networks open source Quagga package helped bring routing capabilities on the host, giving application mobility and scalability.

- Provided full bandwidth via ECMP in layer 3 with multiple spine switches instead of traditional layer 2 topologies with VLANs, which can be functionally limiting.

- Utilized standard Linux command line tools for debugging and troubleshooting such as ping, traceroute, ethtool, ifconfig, and tcpdump.