DataCenter Project Implementation

People

- Principle Cloud Solutions Architect
- Enterprise Architect
- Network Architect
- Head of R&D

Software Defined Data Center

An architecture that is independent of the hardware present - Compute,Storage,Networking,Environmental - Virtualized storage,networking - Automated Operations - SDDC is hardware agnostic

Physical Hardware

- Compute
- Networking
- Storage

Benefits of SDDC

- Agility [autoscaling s]
- Resilience [proper design - near zero-downtime]
- Standardization [great ease of management and scalability]

OpenStack as SDDC

- Cloud Operating System
- Basic Components(neutron,. . . .)
- Future Data Center - API based

Project Methodology

1. Delivery Framework

- Discover
- Design
- Deploy

2. Sprints:

- Hardware assurance/Infrastructure validation with Redhat EL
- Basic Deployment Basic OpenStack API
- Adding Ceph,Swift,SSL,etc. For object storage
- Adding post-deployment automatic tasks Custom Scripts
- Running post-deployment tests Creating a basic tenant

Sample Architecture

1. 3 Controllers,
2. 12 Computes,
3. 5 Ceph OSD
4. 3 Swift Object Storage Nodes
5. VLAN/VXLAN Networks

Other Aspects

1. Centralized Logging - **fluentd**, **ElasticSearch**, **Kibana**
2. Performance Monitoring - **collectd**, **Graphite**,**Grafana**
3. Ceph Monitoring and Administration - **Calamari**
4. Cloud Management Platforms - **Red Hat CloudForms**

- Manage multiple clouds from a single view
- Self-Service Portal
- Utilize HOT(Heat Orchestration Templates)
- Reporting, Dashboards
- Chargeback
- Scale OpenStack from a dashboard

End-To-End Software Defined

1. Why RedHat OpenStack? **Biggest and mostly experienced contributor to the community**
2. RedHat is the market standard
3. With Novatta you get: Wide Infrastructure Knowledge and skills, Professional services and support.

Openstack Private Cloud

- Tenants (Line of Business)
- App Developers
- IT operations

- Private Cloud

Load Balancer as a Service Requirements and Challenges

Full-featured Load Balancer with Operational Simplicity 1. Self-service,fully automated provisioning through *Horizon*, *REST API*, CLI 2. Enterprise scale features: - HA,SSL,multi-tenancy integrated with **Keystone** 3. Per-tenant isolation of load balancer instances 4. Integration with automation tools(*Puppet*, *Ansible*, *Heat*)

Questions we want to answer

Cumulus OS that runs on switches 1. Quicky deploy the compute nodes? 2. Prototype the entire pod virtually first 3. The entire pod be entirely Layer 3 with VXLAN 4. **Ansible** and **Git** be the common language between Network Engineers and System Administrators 5. RHOSP Director Handle the Bulk deployment 6. Do the entire project remotely with ease

Step 0 : Find a Lab/Center

Project Inventory

1. Overcloud

- 303 Dell R220 1U servers (compute,controllers) 16 GB RAM, 2 Gigabit ethernet Ports,Quad core processors
- One Dell R360 server (undercloud)
- 9 Physical Racks

2. Network

- 6 Dell S6000 switches (spine)
- 18 Dell s4048 switches (leaf)
- Cumulus Linux OS
- Cumulus Quagga Linux Package

3. **OpenStack Distro** : Red Hat OpenStack Platform

Step 1: Design the Network (Cumulus)

1. Layer 3 networking throughout with Cumulus Liux
2. Scalability using Ansible and Git - Linux all the way down
3. Compute deployment simplicity with Cumulus Quagga Linux Package

4. Dell Open networking Switches with ONIE bootloader

Step 2: Build the Virtual Prototype (RedHat)

- Build and provision spine/leafs using Cumulus VX and Ansible
- Deploy 5 Compute Nodes using Redhat OSP-d
- Build Ansible Inventory file from nova list bash script
- Bootstrap compute nodes by building management network
- Run Ansible site playbooks to install Cumulus Quagga and configure nodes to join L3 fabric

Step 3: Deploy to Physical (Cumulus/RedHat)

- Reuse all Ansible scripts from virtual and apply to physical
- Deploy Overcloud on 60 Dell R220 node *batches* due to undercloud and controller hardware configuration
- Stop at 300 compute nodes, and 3 controllers
- Create 1000 tenant networks across compute nodes

Step 4: Analyze Results

- 1000% Linux in the entire rack
- Built and provisioned the network in 15 minutes without proprietary APIs or controllers All Layer 3 network using BBGP Unnumbered, no VLANS(VXLAN), and reduced IP address bookkeeping
- Built overcloud in less than 6 hours with hardware provided
- Network and overcloud deployed only with Ansible,Git,ZTP and RedHat OpenStack Project Director
- Stress test with Rally and analyze with Browbeat