Tungsten Fabric Microservice Architecture & Role in Edge Computing

Qasim Arham
Nov 6, 2018 (OSN Day Dallas)
COMMUNITY MEMBERS

CAVUIM
Lenovo
codilime
NTT Innovation Institute Inc.
Riot Games
CloudOps
TELUS
Workday
cloud dynamics
Intel
Aricent
CloudSeeds
Ubuntu
IBM
ebay classifieds group
AT&T
Juniper Networks
redhat
CertusNet
SERRO
Inmarsat
Symantec
SCALR
SDN Essentials
Mellanox Technologies
MIRANTIS
Netronome
Shenzhen Stock Exchange
Bell
Yandex
orange Business Services
IPnet
Semihalf
techtrueup
Atos
STC
Jio
Reliance Communications
CENGN
CERN
FUGA CLOUD
Wingu

your logo here
Networking for Edge Computing

Networking is most overlooked and underestimated component in any stack

Networking is focal point for most of the security and scalability issues

Tungsten Fabric is fully distributed and Microservices based SDN controller addressing security, scale and advance networking services

Production grade networking stack for Data Center and Public & Edge cloud

Highly available and ISSU (In Service Software Upgrade) support

Full Fabric Management – Overlay & Underlay Networks
Tungsten Fabric as SDN Controller

RULE THEM ALL WITH ONE
automated secure open SDN Controller

Public & Private IaaS

CaaS & PaaS

VMs or Metal

openstack

MESOS

OPENSHIFT

kubernetes

VMware

KVM
Architecture Overview
Visualizing Tungsten Fabric’s Operational Effects

TF Security Policy (e.g. allow only HTTP traffic)

Service Chain Policy with a Firewall VNF

Intra-network traffic

Inter-network traffic traversing a service

IP fabric (switch underlay)
Tungsten Fabric Multi Cloud
Multi-Cloud Networking for Converged Operators

Single Pane of Glass for Multi-Cloud Network Management
Distributed Policy Orchestration
Fabric Deployment and Lifecycle Automation

Tiers (i.e. collection of workloads) extend across different DC’s and to the public Cloud
vRouter Agent
- Exchanging control state such as routes with the Control nodes using XMPP.
- Receiving low-level configuration state such as routing instances and forwarding policy from the Control nodes using XMPP.
- Reporting analytics state such as logs, statistics, and events to the analytics nodes.
- Installing forwarding state into the forwarding plane.
- Discovering the existence and attributes of VMs in cooperation with the Nova agent.
- Applying forwarding policy for the first packet of each new flow and installing a flow entry in the flow table of the forwarding plane.
- Proxying DHCP, ARP, DNS.

vRouter Kernel/DPDK
- Encapsulating packets sent from the overlay network and decapsulating packets received for the overlay network.
- Packets received from the overlay network are assigned to a routing instance based on the MPLS label or Virtual Network Identifier (VNI).
- Doing a lookup of the destination address of the in the Forwarding Information Base (FIB) and forwarding the packet to the correct destination. The routes may be layer-3 IP prefixes or layer-2 MAC addresses.
- Doing RPF check before sending Virtual machine traffic to destination. This is configurable.
TF VROUTER DEPLOYMENT MODELS

**KERNEL VROUTER**
- This is the normal operation where the forwarding plane of vRouter runs in the kernel and is connected to VMs using TAP interface (or veth pair for containers).
- vRouter itself is enhanced using other performance-related features:
  - TSO / LRO
  - Multi-Q Virtio

**DPDK VROUTER**
- vRouter runs as a user space process and uses DPDK for fast path Packet I/O.
- Full set of SDN Capabilities Supported
- Requires the VMs to have DPDK enabled for performance benefits

**SRIOV/ VROUTER COEXISTENCE**
- Some workloads can directly SRIOV into the NIC, while others go through the vRouter.
- Sometimes a VNF can have multiple interfaces some of which are SRIOV-ed to the NIC.
- Interfaces that are SRIOV-ed into NIC don’t get the benefits/features of vRouter.

**SMARTNIC VROUTER**
- vRouter forwarding plane runs within the NIC.
- Workloads are SRIOV-connected to the NIC.
Tungsten Fabric Evolution to Microservices

- **Contrail-Control (5 daemons)**
- **Contrail-Config (8 daemons)**
- **Contrail-Analytics (5 daemons)**
- **Contrail-WebUI (4 daemons)**
- **Contrail-DB (3 daemons)**
- **Contrail-vRouter (3 D) + Kernel/DPDK (FP)**

**Contrail Controller: 2n+1**

- VM
- VM
- VM

**OR**

- BMS

**Contrail 1.X/2.X/3.X**

BMS or VMs base (SDN Controller)

---

**Multiple Process running in one Container (FAT Containers)**

- Analytics
- DB
- Config + Control
- Analytics
- Kube MGR
- HA Proxy
- vRouter Agent

**DaemonSet, Ingress Services with Host Networking**

with choice of run single or multiple containers per PODs

- 27-30 Containers Images

**Contrail 4.X (Containers)**

BMS or VMs base (SDN Controller)

**Contrail 5.X (Containers)**

Microservices (SDN Controller)
TF Helm Microservices Architecture *(Helm Charts)*

---

**POD – DaemonSet (3/3)**
- Container (1/4): contrail-control-nodemgr
  - Host-Networking
- Container (2/4): contrail-dns
  - Host-Networking
- Container (3/4): contrail-named
  - Host-Networking

**POD – DaemonSet (5/5)**
- Container (1/5): contrail-config-api
  - Host-Networking
- Container (2/5): contrail-config-nodemgr
  - Host-Networking
- Container (3/5): contrail-svc-monitor
  - Host-Networking
- Container (4/5): contrail-schema-transform
  - Host-Networking
- Container (5/5): contrail-device-mgr
  - Host-Networking

**POD – DaemonSet (2/2)**
- Container (1/2): contrail-webui
  - Host-Networking
- Container (2/2): contrail-webui-middleware
  - Host-Networking

**POD – DaemonSet (6/7)**
- Container (1/7): contrail-analytics-api
  - Host-Networking
- Container (2/7): contrail-analytics-nodemgr
  - Host-Networking
- Container (3/7): contrail-collector
  - Host-Networking
- Container (4/7): contrail-snmp-collector
  - Host-Networking
- Container (5/7): contrail-query-engine
  - Host-Networking
- Container (6/7): Contrail-topology
  - Host-Networking
- Container (7/7): Contrail-alarm-gen
  - Host-Networking

**POD – DaemonSet (7/7)**
- Container (1/1): contrail-control
  - Host-Networking
- Container (2/1): contrail-dns
  - Host-Networking
- Container (3/1): contrail-named
  - Host-Networking

**Kubernetes Cluster**
- contrail-config
- contrail-control
- contrail-webui
- contrail-analytics
- contrail-vrouter

**Other Containers:**
- Contrail-status
- node-init
- vrouter-init-kernel
- vrouter-init-dpdk

**Contrail Helm Toolkit**

---

**Contrail-Controller**

**Contrail-Analytics**

**Contrail-Third-Party**

**POD DaemonSet (2/2)**
- Container (1/3): contrail-vrouter-agent-dpdk
  - Host-Networking
- Container (2/3): contrail-vrouter-dpdk
  - Host-Networking
- Container (3/3): contrail-vrouter-nodemgr
  - Host-Networking
Tungsten Fabric Integration with ONAP

Orchestration Config REST API (MSO)

E2E monitoring Analytics REST API (DCAE)
Edge Computing (Today & Tomorrow)

**Today**
- Cell Sites
- Core Sites
- Internet

**Tomorrow**
- Cell/Edge Sites
- Core Sites
- Internet
- Colo/Internet
- Distributed Core Sites
- BMS, VM, Containers
- Colocation Data Center

Additional Info:
- AWS
- Google Cloud
- Packet
- On Premise
TF Distributed Compute Architecture

Core/Distributed Core Site
Distributed Compute (Sub-Cluster-ID)

- TF SDN Controller (Cluster)
- Sub-Cluster Controller (1)
- Sub-Cluster Controller (n)
- Local Compute

IP/MPLS
Backbone/RAN Transport

Light version: BGP not extended to Edge sites only; XMPP to Edge Sites

Core/Distributed Core Site
Distributed Compute Light (Sub-Cluster-ID)

- TF SDN Controller (Cluster)
- Sub-Cluster Controller (1)
- Sub-Cluster Controller (n)
- Local Compute

IP/MPLS
Backbone/RAN Transport

Edge Site (1)

Edge Site (n)

Edge Site (1)

Edge Site (n)
TF Distributed Compute Architecture

Core/Distributed Core Site
Distributed Compute (Sub-Cluster-ID) with MC-GW

- Contrail SDN Controller (Cluster)
- Sub-Cluster Controller (1)
- Sub-Cluster Controller (n)

Local Compute

Distributed Compute

Edge Site (1)

Edge Site (n)

Colo/Internet/IP/MPLS
Backbone/RAN Transport

IPsec/SSL
XMPP
Multi-Cloud Gateway
TF as Single SDN for VMs, PODs & BMS

**Basic Networking:**
L2/L3 or L2/L3 Network, IPAM/DHCP, DNS, Multi-Tenancy

**Advance Networking:**
VLAN-ID, VRRP, VIP, Load Balancer, Routes Advertisement, GW Function, Service Chaining, Traffic Steering, Flow awareness, QoS, vRouter Kernel/DPDK, SR-IOV, BGP-VPN, Inter Site Federation DCI, Health Checks, FW, IPSec/SSL Support, Distributed Compute, Edge Fabric Management, Multi-Cloud support, Multi-tenancy (to support network slicing)
Secure RAN to CN
- Use Contrail Encryption to secure Remote Edge and Central DC connection.
- Secure Overlay site to site communication via Contrail encryption support
- Policy based encryption model
Tungsten fabric provides a rich, consistent set of security policy capabilities across multiple platforms.

1. Simplified Manageability *(change control, etc. is much easier)*
2. Improved Scalability
3. Define / Review / Approve Once → Use Everywhere
Tungsten Fabric INSTALLATION

- Ansible playbook to flexibly deploy Tungsten Fabric binaries
- Helm charts to easily operate Tungsten Fabric components on Kubernetes
- Install-time option with OpenShift to deploy with Tungsten Fabric
- Tungsten Fabric binaries available on DockerHub and we're improving CI/CD
- Commercial integrations into lifecycle tools like RH OpenStack Director
Tungsten Fabric K8s CNI (A single YAML Install & CARBIDE)

Reference: https://github.com/Juniper/contrail-kubernetes-docs
Carbide Sandbox Environment

Tungsten Fabric + Kubernetes on AWS

https://tungsten.io/start/
**0-60 in 15 Minutes Flat w/Carbide (TF+k8s on AWS)**

<table>
<thead>
<tr>
<th>Stack Name</th>
<th>Created Time</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MasterNodeURL</td>
<td><a href="https://ec2-54-183-143-157.us-west-1.compute.amazonaws.com">https://ec2-54-183-143-157.us-west-1.compute.amazonaws.com</a></td>
<td>URL for Master node of Sandbox (u...</td>
</tr>
</tbody>
</table>
0-60 in 15 Minutes Flat w/Carbide (TF+k8s on AWS)

Deployment is in progress:
Please wait until the deployment ends.

23:27:46 UTC: 1/6 The control site is being deployed ...
23:28:45 UTC: 2/6 Creating and exporting a key pair ...
23:28:46 UTC: 3/6 Download the repository ...
23:28:46 UTC: 4/6 Provision instances ...
23:31:02 UTC: 5/6 Configure instances ...
23:33:52 UTC: 6/6 Install Kubernetes and Tungsten Fabric ...
Deployment is completed

Contrail UI: https://ec2-54-215-222-33.us-west-1.compute.amazonaws.com:8143

User name: admin
User password: contrail123

To use Tungsten Fabric or Kubernetes command line utilities, connect to controller using the key specified during the deployment of CloudFormation stack and centos user name.

Example:

    ssh -i randy-b-carbide-test.pem centos@ec2-54-215-222-33.us-west-1.compute.amazonaws.com

Accessing the Kubernetes dashboard:

On the controller:

    kubectl get pods -n kube-system -o wide | grep dashboard

Check the IP column. It tells you the private IP address of the compute node where the dashboard POD is running. You need to find out the associated public IP address (it is left to you as an exercise). Once you know it, you can connect to the URL:

    https://<public-ip>:8443

Select the token option. Where can you get the token from? There is one on the controller’s file /root/k8s_dashboard_token.txt, but it only allows to visualize. If you want read-write access do the following:

    kubectl get secret -n contrail | grep kubemanager
    kubectl describe secret -n contrail | grep "token:" | awk '{print $2}'

Take your time to browse the dashboard. During the next exercises, you can choose to do some tasks on the web instead of (or in addition to) the CLI.
Carbide EC2 Instances overview

- tungstenfabric-k8s-aws-master-node
- tungstenfabric-k8s-aws_control1
- tungstenfabric-k8s-aws_compute1
- tungstenfabric-k8s-aws_compute2
0-60 in 15 Minutes Flat w/Carbide (TF+k8s on AWS)
Try Tungsten Fabric
