AGENDA

• The state of cloud and virtualisation
• Operating in a multicloud world
• Juniper’s Contrail Enterprise Multicloud
“CLOUD” IS NOT JUST A LOCATION

**Multiple IT Platforms**
- vendors, regions, AZs, accounts, virtualization stacks...

**Cloud IT Platform**
- Cohesive management

**DISJOINT, LEGACY, SILOED ENVIRONMENT**
- Locked into each silo
- No portability of apps
- Data is not normalized
- Security is inconsistent and uneven
- Network policy is bespoke
- WAN transport is expensive and reach is contained

**CLOUD IT PLATFORM**
- Flexibility of best venue and economics
- Portability of apps
- Data is normalized and access is standardized
- Security is visibility, spans boundaries
- Secure network policy must spans boundaries
- WAN must be secure and optimized in overlay and underlay
WHERE DID WE COME FROM?

• Virtualization went a long way to allow increasing capability and efficiency of compute hardware to be leveraged whilst supporting current application architectures.

• In addition, hyper-convergence (collapse of storage and compute) further removes the specialization of hardware to application matching.

• Web scalers brought the concept of full application and storage elasticity, making horizontal scaling, disaggregated from hardware, infinite, with the added benefit of resiliency.
  • MapReduce, Hadoop, etc
  • Tiered load balance, web front end, service backend, storage domains for independent scaling.

But, this flexibility was initially embodied through custom architectures and development tools.
FROM VIRTUALIZED TO CONTAINERIZED

• Allow packaging of applications into functions
• Mostly breaks the decades long hold of Operating System dependency, and far removed from hardware
• Containers can be scaled independently at each layer of application architecture (front end, back end, etc)
• This further enables financial scalability too; on-demand, spot pricing, dynamic baselining
• Can be created and destroyed dynamically, allowing continuity of operation across load, maintenance and location – worker bees in a hive

© 2019 Juniper Networks

Google Kubernetes Engine

AWS Lambda

Google Kubernetes Engine
CONTAINER ADOPTION IS RAPID

Organization’s adoption plans for containers (e.g., Docker)

- In use: 19%
- In discovery/proof of concept: 19%
- Plan to implement in next 12 months: 7%
- Plan to implement in next 24 months: 3%
- Considering, but no current plan to implement: 21%
- Not in-use/not in plan: 31%

% of respondents (n=426)

Q. Please indicate your organization’s adoption plans for containers (e.g., Docker).
Source: 451 Research, Voice of the Enterprise: Servers and Converged Infrastructure, Workloads and Key Projects 2018
FROM CLOUD TO MULTICLOUD

• The facilities enabling virtualisation are no longer single-vendor or closed ecosystems
• Further more, business needs demand multiple platforms and locations to ensure reliable infrastructure and a vibrant marketplace
COMBINED: MICROSERVICES & MICROSEGMENTATION

• Microservices and containerization work hand in hand to allow application and business functions to be segmented for scale, resilience and development independence – a direct embodiment of DevOps in terms of application architecture

• Micro-segmentation is then the segmentation of network functions (primarily security) to support the coexistence of many microservices within a single networking domain
  • May be by application tier (front-end, back-end, etc), or;
  • By delivery environment (development, staging, production), or;
  • Both, such as in a multi-tenant environment

So why take this approach?
Agility & Impact

Diverse & Decoupled
MULTICLOUD CONSIDERATIONS

Advantages

• Simplify OS environment
• More efficiently leverage resources (compute, storage, network)
• Portability, right size location and cost
• Development agility – reduce interdependence and adopt microservices principles

Challenges

• Reduce visibility and control of host environment, particularly networking & security
• Diverse and duplicate set of applications in every physical/network location increases costs
• Indeterminate physical/network location, difficult accounting
• Troubleshooting issues between components is now a “network” problem
EXAMPLE ENVIRONMENT – 3 TIERED APPLICATION

Environment A (e.g. Public Cloud)

Environment B (e.g. Private/Hosted Cloud)

Compute Node

Compute Node

Compute Node

Compute Node
THE CHALLENGE: MANAGING SECURITY ACROSS ENVIRONMENTS

VIRTUAL NETWORK GREEN

Security Groups
G1, G2, G3

Intra-network traffic

Inter Virtual Network Policy (e.g., allow only application traffic)

Allow internal application traffic only

VIRTUAL NETWORK BLUE

Service chaining Layer 7 / IPS Firewall

B1, B2, B3

Inter-network traffic traversing a L7 Firewall

VIRTUAL NETWORK YELLOW

VM and virtualized Network function pool

Y1, Y2, Y3

Intra-network traffic

Inter-network traffic

PHYSICAL (Policy Enforcement)

Host + Hypervisor

G1, B3, Y1, G3, B1

LOGICAL (Policy Definition)

PHYSICAL (Policy Enforcement)

Host + Hypervisor

G2, B2, Y3, Y2

IP Fabric (switch underlay)
**POLICY CHALLENGES**

**Current Behavior**

- App1, Deployment = Dev
  - Network Policy = P1

- App1, Deployment = Staging
  - Network Policy = P2

- App1, Deployment = Prod
  - Network Policy = P3

**Desired Behavior**

Can we use one policy to be applied in all the different deployments?

1. **Reduced Complexity** (fewer policies)
2. **Simplified Manageability** (change control, etc. is much easier)
3. **Improved Scalability**

...
POLICY OPTIMIZATION
WRITE ONCE – DEPLOY MANY

Once a set of policies are defined for a particular environment, they can easily be re-used for other environments.

1. **Reduced Complexity** (less # of policies)
2. **Simplified Manageability** (change control, etc. is much easier)
3. **Improved Scalability**
4. **Define / Review / Approve Once → Use Everywhere**

- **Policy**
  - **App1, Deployment = Dev** (Web, App, db)
  - **App1, Deployment = Staging** (Web, App, db)
  - **App1, Deployment = Prod** (Web, App, db)
  - **Bare Metal Servers** (App1, Deployment = Staging-BMS)
  - **App1, Deployment = Dev-AWS** (Web, App, db)
  - **App1, Deployment = Dev-K8s** (Web, App, db)
  - **App1, Deployment = Dev-OpenShift** (Web, App, db)

No policy rewrite needed:
- **WebApp db**
- **App1, Deployment = Dev**
- **App1, Deployment = Staging**
- **App1, Deployment = Prod**
- **App1, Deployment = Staging-BMS**
- **App1, Deployment = Dev-AWS**
- **App1, Deployment = Dev-K8s**
- **App1, Deployment = Dev-OpenShift**
CONTRAIL ENTERPRISE MULTICLOUD

Contrail Enterprise Multicloud

Any Location
- Private cloud, data centers, public cloud

Any Workload
- Bare metal servers, public cloud instances, virtual machines, containers and physical networking devices

Any Deployment
- Greenfield or brownfield, single- or multi-vendor

One, open platform for end-to-end policy and control with analytics
USE CASE: MULTICLOUD - PRIVATE TO PUBLIC CLOUD

One Platform for All Clouds
• Connect multiple virtual networks across data centers and public clouds
• Monitor and control network and security policy for workloads anywhere on the network
• Securely connect bare metal servers with VMs and containers across private and public clouds
• Overlay networking services between cloud instances
• Supports distributed application architectures

Manage workloads in multiple clouds as though they were in one
USE CASE: APPLICATION SECURITY W/ MICROSEGMENTATION

Secure Applications
- Configure and apply fine-grained security policy to workloads on any compute
- Enforce security policy with distributed L4 firewalls
- Isolate workloads and tenants while sharing the cloud resources
- Redirect traffic to a L7 firewall for extra protection
- Enhance K8s networking service to provide high performance connectivity and isolation where needed
- Support multiple K8s deployment types - K8s on BMS, OpenStack, Public Clouds

Consistent security for multiple environments, including Kubernetes
HOW?
COMPUTE NODE: VROUTER ON HYPERVISOR

- vRouter replaces the Linux Bridge or OVS module in Hypervisor Kernel
- vRouter performs bridging (EVPN) and routing (L3VPN)
- vRouter performs networking services like Security Policies, NAT, Multicast, Mirroring, and Load Balancing
- Replaces need for Service Nodes or L2/L3 Gateways for Routing, Broadcast/Multicast, NAT
- Routes are automatically leaked into VRFs based on Policies
- Support for Multiple Interfaces on the Virtual Machines
- Support for Multiple Interfaces from Compute Node to the Switching Fabric
EXAMPLE ENVIRONMENT – 3 TIER APP, MULTICLOUD

End Users

Web Tier

App Tier

DB Tier

Compute Node

WAN

Compute Node

Compute Node

WAN

Compute Node

Compute Node

Private Cloud

kubernetes
CONTRAIL IN ACTION – WORKLOAD VISIBILITY

Centralised view of resources and workloads no matter where they reside
CONTRAIL IN ACTION – POLICY CREATION

Central editing of policy with easy to understand tagging, common across domains
CONTRAIL IN ACTION – POLICY VISUALISATION

Visualisation of flows and action by domain...
CONTRAIL IN ACTION – MICROSEGMENTATION VISUALISATION

...with the ability to drill down to workload location for true microsegmentation visibility
**ENCRYPTED MULTICLOUD OVERLAY**

End-to-End Encryption provides a seamless fabric between heterogeneous compute environments

Encryption of Data Plane (vRouter to vRouter) within the private cloud

1. **Group-based VPN (IPsec mesh to vRouter) or SSLVN (with dTLS)**
2. **L4 Policy based encryption**

... And encryption extended to Public Cloud

- Internet or Direct Connect
- **vRouter**
- **vRouter**
- **GW**
- **VPC**
• **Business Challenge:** Supporting hosted classified advertising platforms in multiple countries around the world, eBay Classifieds must meet requirements unique to each market with very little functional overlap

• **Solution:** Leveraged OpenStack features using Juniper Network’s Contrail Cloud Platform and Professional Services
  • Plus Juniper Networks QFX5100 based IP Fabric Underlay

• **Benefits:** Avoided expensive vendor-lock-in and enabled interoperation with numerous hypervisors, orchestration systems and physical networks
  • Delivers flexible automated network design that adapts to users’ needs

• **Why SDN:** “We also know that having a consistent API as an abstraction layer above the hardware is a key requirement for Riot to effectively scale its network configuration management and operations. In our time of need, we turned to overlay networks.”

  • “We landed on [OpenContrail](https://www.opencontrail.io) from Juniper Networks. [Contrail] is designed from the ground up to be an open source, vendor-agnostic solution that works with any existing network. BGP and MPLS are at its core - both are known protocols that have been proven to scale to the size of the entire internet.”

  • “Engineers no longer have to ask questions like “how many ports does this have”, “what vendor are we using,” or “where should the security policy live?” Instead, we can deliver a consistent API to program what the engineers do want to concern themselves with. ...by having the same API available in every data center that Riot operates, we can write automation that works everywhere, every time...”
WHERE TO FROM HERE?

• Contrail is here to manage and visualise networking and security across on-prem and cloud environments, proven integration with:
  • Nutanix
  • VMWare
  • OpenStack
  • Kubernetes
  • Public clouds (AWS, Azure, GCP)

• It is built ready for the microsegmentation and containerisation trend

• Can also manage your Juniper data centre underlay (Contrail Fabric Management)

Let us engage with your use case and environment and show you how we can unify them under Contrail Enterprise Multicloud