Wie Microservices die Anforderungen an Netze neu definieren

Thomas Kretzschmar
VMware
tKretzschmar@VMware.com
Januar 2020
Evolution of Infrastructure

1995

Virtualization Wave 1: Workload Consolidation

Client Server

Virtualization Wave 2: Workload Mobility

Mainframe/Midrange

Compute

Present

Cloud Delivery
Evolution of Infrastructure

1995 - Present

Compute:
- Mainframe/Midrange
- Client/Server
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery

Network:
- Convergence > Ethernet and IP
- Feeds and Speeds and Reliability

Virtualization Wave 1:
- Workload Consolidation

Virtualization Wave 2:
- Workload Mobility
Evolution of Infrastructure

- **DAS / SAN / NAS**
  - Convergence > Ethernet and IP
  - Feeds, Speeds, and Capacity

- **Network**
  - 1995: Mainframe / Midrange
  - Client Server
  - Virtualization Wave 1: Workload Consolidation
  - Virtualization Wave 2: Workload Mobility
  - Cloud Delivery
  - Feeds and Speeds and Reliability

- **Compute**
  - 1995: Mainframe / Midrange
  - Client Server
  - Virtualization Wave 1: Workload Consolidation
  - Virtualization Wave 2: Workload Mobility
  - Cloud Delivery

- **Present**
  - Virtualization Wave 2: Workload Mobility
  - Cloud Delivery
Evolution of Infrastructure

Compute
- Mainframe
- Client Server
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery

Network
- Convergence > Ethernet and IP
- Feeds and Speeds and Reliability

Storage
- DAS / SAN / NAS
- Feeds, Speeds, and Capacity

Security
- Firewalls and ACLs
- More Firewalls and ACLs
- Even More Firewalls and ACLs

1995
- Client Server
- Mainframe/Midrange

Present
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery

Virtualization Wave 1: Workload Consolidation
- Mainframe
- Client Server
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery

Virtualization Wave 2: Workload Mobility
- DAS / SAN / NAS
- Feeds, Speeds, and Capacity
- Firewalls and ACLs
- More Firewalls and ACLs
- Even More Firewalls and ACLs

Cloud Delivery
- DAS / SAN / NAS
- Virtualization Wave 2: Workload Mobility

Firewalls and ACLs
- Mainframe/Midrange
- Client Server
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery

More Firewalls and ACLs
- Mainframe/Midrange
- Client Server
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery

Even More Firewalls and ACLs
- Mainframe/Midrange
- Client Server
- Virtualization Wave 1: Workload Consolidation
- Virtualization Wave 2: Workload Mobility
- Cloud Delivery
Monolith Apps Running on Monolith Infrastructure

Communicate between different Apps. Networking as usual

- **Complexity** – As it grows, can be difficult to understand
- **Anti-Agility** – Long release cycles and change windows
- **Availability** – Bug can bring down the entire application
- **Technology** – Long-term commitment to a stack
Microservices Running on “Microinfrastructure”

Communicate between different microservices. Networking as usual?

**Agility** – Innovation and business value

**Elasticity** – Horizontally and independently

**Resiliency** – Faults are quickly isolated

**Technology** – Adopt new frameworks and languages
Apps are Changing

Monolithic Development: Long cycle

Then
- IT time scale
- On-prem / Perimeters

Frequency of deployment

Now
- Dev time scale
- Multi-cloud / Borderless

Legacy Apps

Cloud native apps

Size of Change

High touch / High value
- Perceived to be understood
- Risk analysis / Preventive measures

Low touch / Difficult to assess value
- Not understood
- Risk what??
State of the Market: Monoliths to Microservices Migration

Monolith

Front-end

Business Logic
(Auth, Catalog, Order, Shipping)

Data Interface

DB

Microservice Architecture

Front-end

Microservice (Auth)

Microservice (Catalog)

Microservice (Payment)

Microservice (Order)

Microservice (Cart)

Microservice (Shipping)

DB

Java

Python

©2019 VMware, Inc.
Scaling Microservices
Scaling Microservices
Scaling Microservices
Scaling Microservices

Diagram showing a network architecture with various components such as Load Balancer, Front-end Webserver in Container, Auth, Payment, Order, Catalog, Cart, and Shipping. Each component is connected to the load balancers and databases (DB) as part of the scalable architecture.
CNCF 2018 Survey – Challenges

- Complexity
- Cultural Changes
- Difficulty Choosing an Orchestration Solution
- Finding Vendor Support
- Lack of Training
- Logging
- Monitoring
- Networking
- Other
- Reliability
- Scaling Deployments
- Security
- Storage

Nov ’16 | Mar ’17 | Dec ’17 | Jul ’18

©2019 VMware, Inc.
What is Networking?

- **Identity**
  - IP, DNS, Certificates

- **Connectivity**
  - Switches and routers

- **Security**
  - Firewall, IPS, VPN, WAF, ...

- **Scale**
  - LB, GSLB
Port Mapping

Managing port mappings is challenging and not used in production by enterprises.

Container Networking Evolution

Container Network Interface (CNI)

Provides a unique IP address per container.
Network Automation for Kubernetes

Model 1: Dedicated tenant routers per K8 Namespace

```
admin@k8s-master:~$ kubectl create namespace foo
namespace "foo" created

admin@k8s-master:~$ kubectl create namespace bar
namespace "bar" created

admin@k8s-master:~$ kubectl run nginx --image=nginx -n foo
deployment "nginx-foo" created

admin@k8s-master:~$ kubectl run nginx --image=nginx -n bar
deployment "nginx-bar" created
```
Container Networking Evolution

Port Mapping

Managing port mappings is challenging and not used in production by enterprises

Host IP A
Port 8001

Host IP A
Port 8002

vSwitch

Container Network Interface (CNI)

Provides a unique IP address per container

Container Net

vSwitch

vSwitch

Service Mesh

Sidecar proxy (e.g., Envoy) container per app enables Layer 7 traffic management and security use cases

Container Net

vSwitch

vSwitch

©2019 VMware, Inc.
Let’s Move Networking and Security Down the Stack

Again!

Client Libraries & App Frameworks

Service Meshes & Sidecars

K8s Pod

App Container

Discovery
Observability
Connectivity
Control
Security

Discovery
Observability
Connectivity
Control
Security
Sidecar
Best Practice for Securing MicroServices

- Use TLS-protocols for all communication
- Use OpenID or OAuth 2.0
- Don’t show sensitive data as plain text
- Protect public APIs from denial-of-service-attacks
- Use encryption before persisting the data
- Automate everything
- Fully map out the communication methodologies between microservices to identify potential problem or irregular behavior.
- Utilize security principles within the code of the microservice
Best Practices for Networking

• Firewalls have to automatically:
  • Creation of new namespaces to segment them from each other
  • Inspect the yaml for security information (tags, ports, ...)
  • react on scaling (up&down)

• At least the user facing service should be secured by a WAF

• Manual configuration of Load Balancers is impossible (Kubernetes does it automatically if the integrated LB is enough)

• Bandwidth is mostly not an issue, but latency quickly multiplies! Test with injected delays!
Additional Links & Hints

Kubernetes directly in ESXi / project Pacific: https://datacenterrookie.wordpress.com/2020/01/06/vmware-project-pacific-making-kubernetes-mainstream/

Project Pacific delivers 8% more performance than Containers on Bare Metal: https://blogs.vmware.com/performance/2019/10/how-does-project-pacific-deliver-8-better-performance-than-bare-metal.html


Thank you!
API FrontEnds

content protection for attacks against API traffic, for example such as CVE-2016-4438: Apache Struts Code Execution.

Rate Limiting protects against denial-of-service attacks, brute-force login attempts, and other types of abusive behavior targeting the application layer.

checking any API call against the profiled API structure to ensure the input parameters (count, order, etc.) are consistent with the definition

Define model API user behavior based on an authentication parameter.
Protection against OWASP top 10 (Injection, Broken Authentication, Cross-Site Scripting).

Differentiate Human and automated traffic (captcha, bots manager)

Application learning to reduce the risk of attacks and helps prevent zero-day vulnerabilities.

Inspects all outbound traffic for sensitive data leakage

Visibility, SIEM integration and log analytics
Thank you!