

Welcome to



June 8, 2017

Virtual machines & containers,
friends or foes?

```
$your_speaker = @{
```

```
name      = 'Yung Chou';  
soundsLike = 'young and chosen';
```

```
company   = 'BLUEMETAL, an Insight Company';  
role      = 'Cloud Architect';  
major     = 'Hybrid Cloud';
```

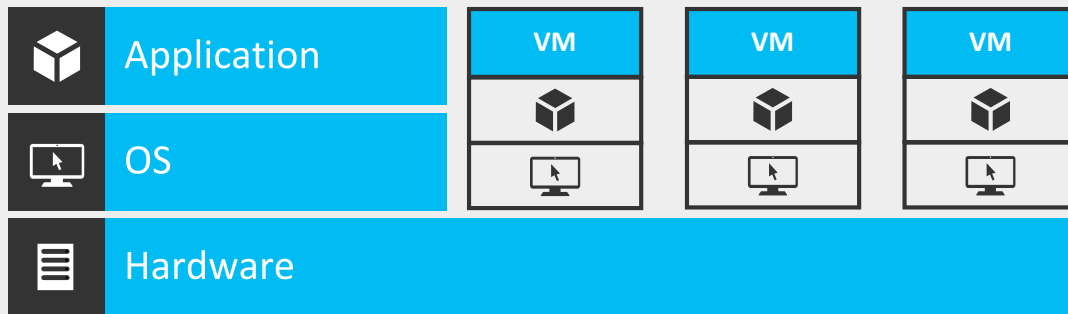
```
blog      = 'http://yungchou.wordpress.com';  
twitter   = '@yungchou'
```

```
}
```



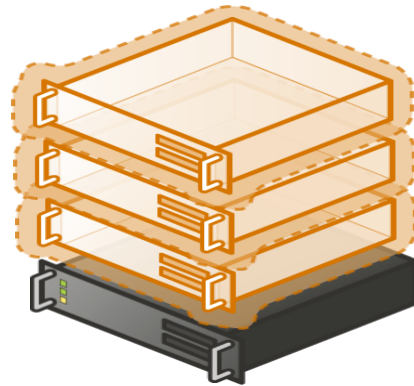
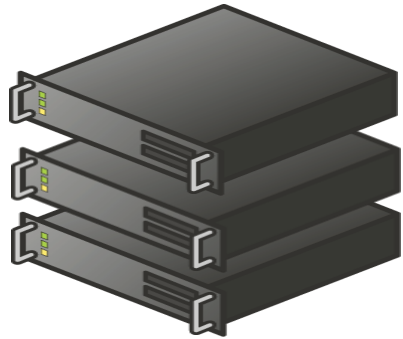
Then

Traditional virtual machines = hardware virtualization

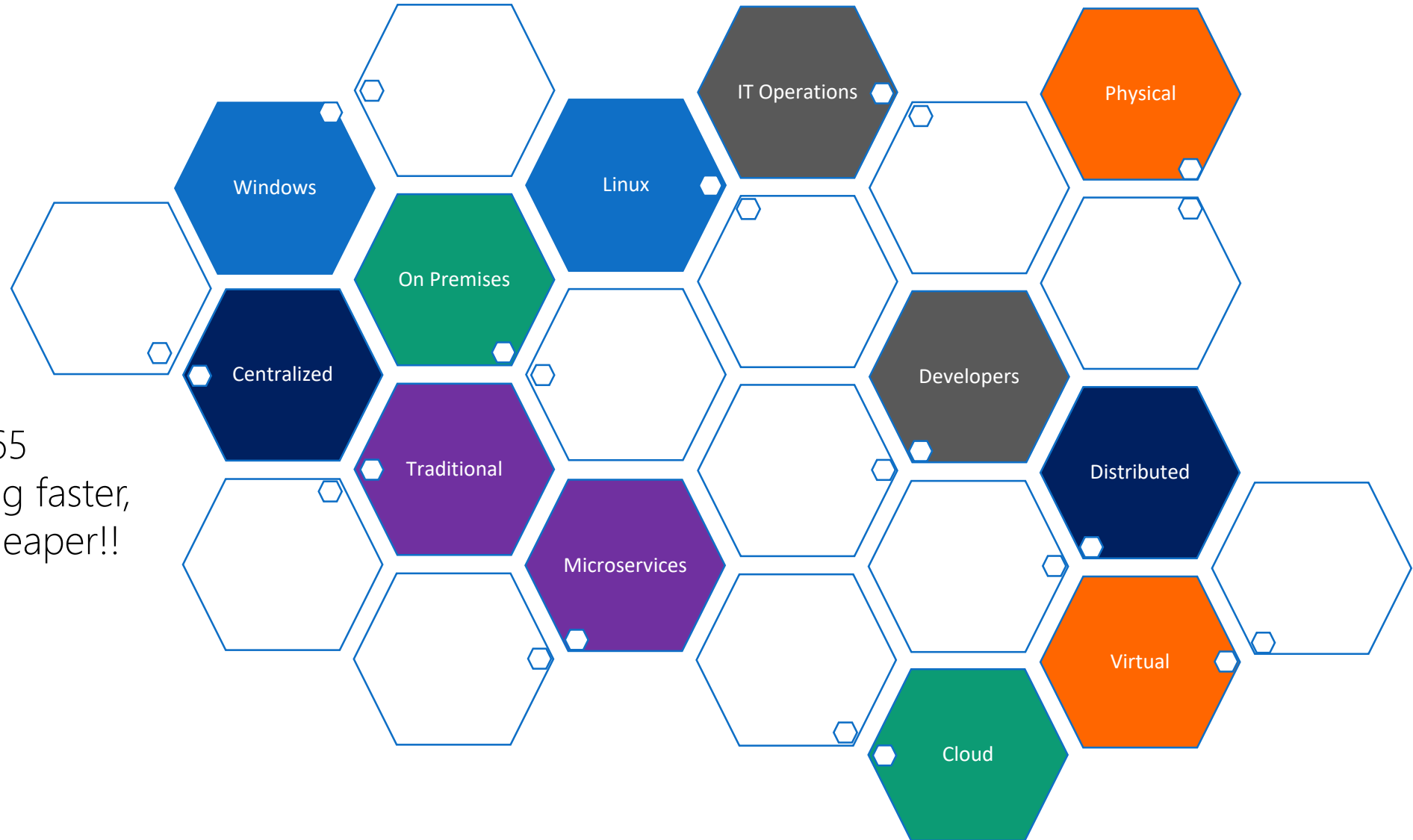


Virtualization

- Hardware virtualization
- Server virtualization
- Server consolidation
- Cost and utilization



Business & technical challenges



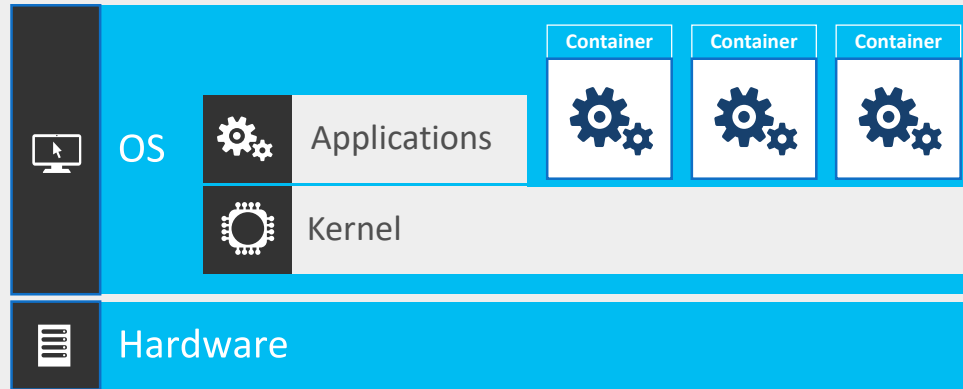
24x7x365
And everything faster,
better and cheaper!!

IT must deliver

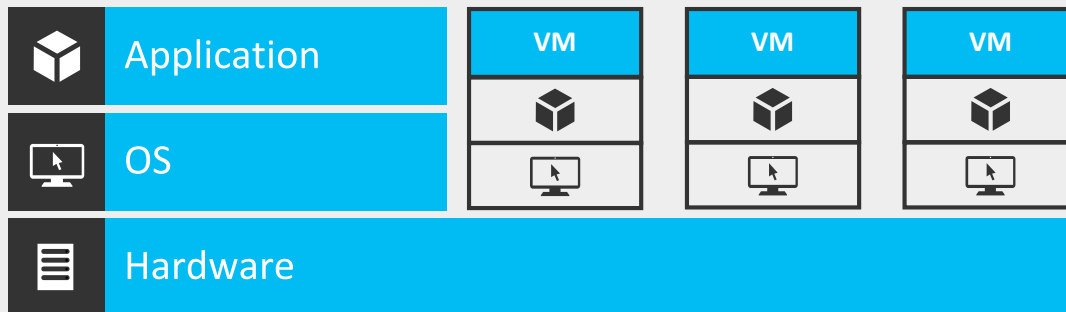
- Agility
 - Easy to set up and use
 - Quick and frequent deliveries
- Portability
 - Storage
 - Network
 - Infrastructure
- Control
 - Manageability
 - Scalability
 - Access and policy

Now

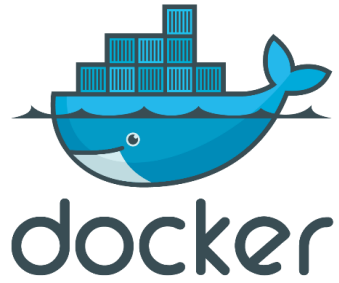
Containers = Operating system virtualization



Traditional virtual machines = hardware virtualization



Demo

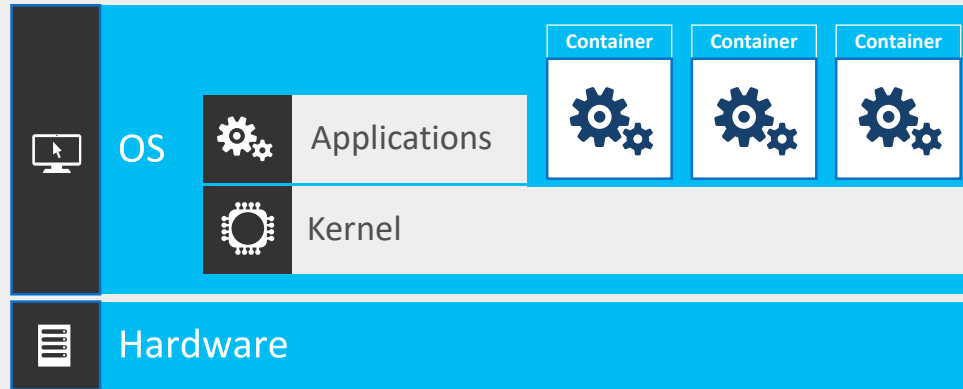


```
Terminal File Edit View Search Terminal Help
ua@u5316:~$ c
ua@u5316:~$ docker version
Client:
 Version:      1.12.1
 API version:  1.24
 Go version:   go1.6.3
 Git commit:   23cf638
 Built:        Thu Aug 18 05:33:38 2016
 OS/Arch:      linux/amd64

Server:
 Version:      1.12.1
 API version:  1.24
 Go version:   go1.6.3
 Git commit:   23cf638
 Built:        Thu Aug 18 05:33:38 2016
 OS/Arch:      linux/amd64
ua@u5316:~$
```

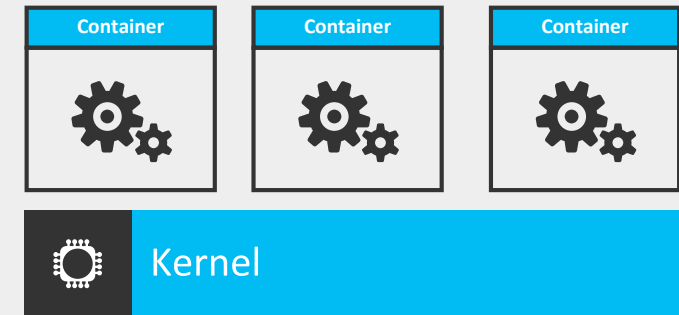
Windows Containers

Containers = Operating system virtualization

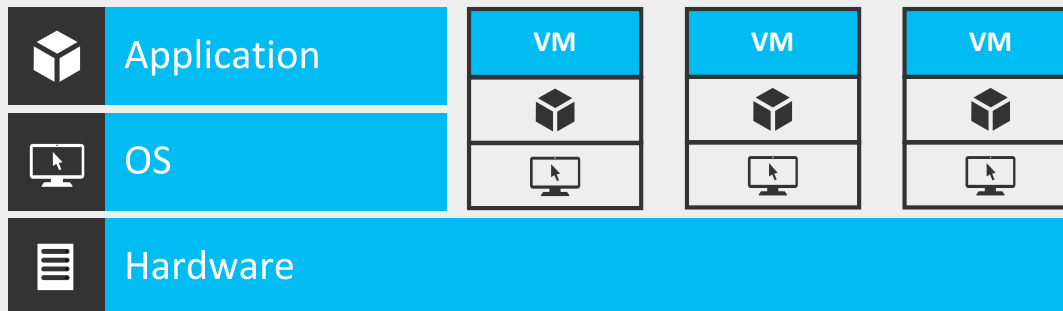


Windows Server containers

Maximum speed and density

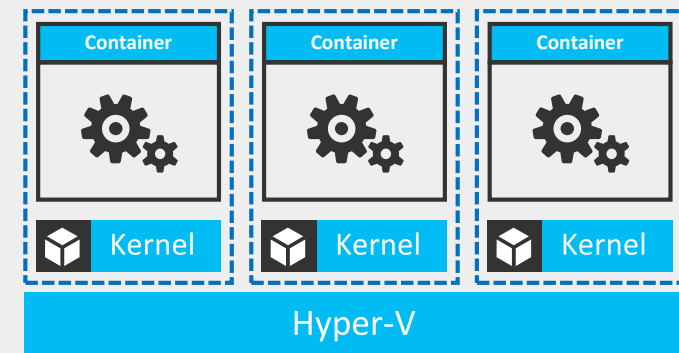


Traditional virtual machines = hardware virtualization

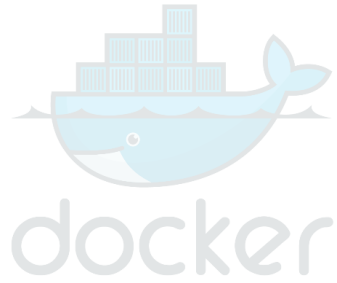


Hyper-V containers

Isolation plus performance



Demo



```
Terminal File Edit View Search Terminal Help
ua@u5316:~$ c
ua@u5316:~$ docker version
Client:
 Version:      1.12.1
 API version:  1.24
 Go version:   go1.6.3
 Git commit:   23cf638
 Built:        Thu Aug 18 05:33:38 2016
 OS/Arch:      linux/amd64

Server:
 Version:      1.12.1
 API version:  1.24
 Go version:   go1.6.3
 Git commit:   23cf638
 Built:        Thu Aug 18 05:33:38 2016
 OS/Arch:      linux/amd64
ua@u5316:~$
```



Windows Server 2016

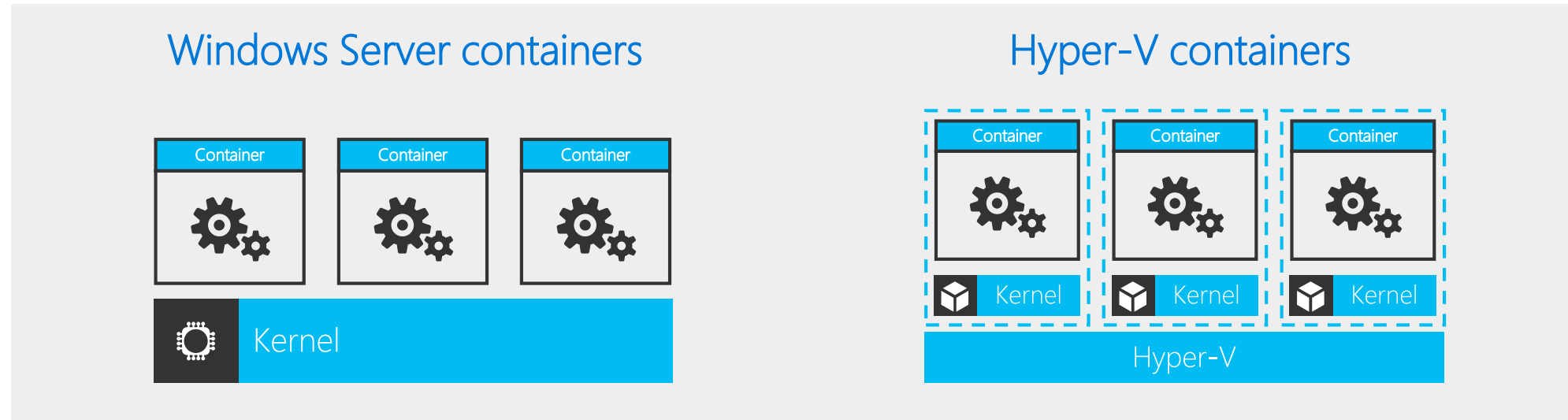


Windows 10



Maximum speed and density

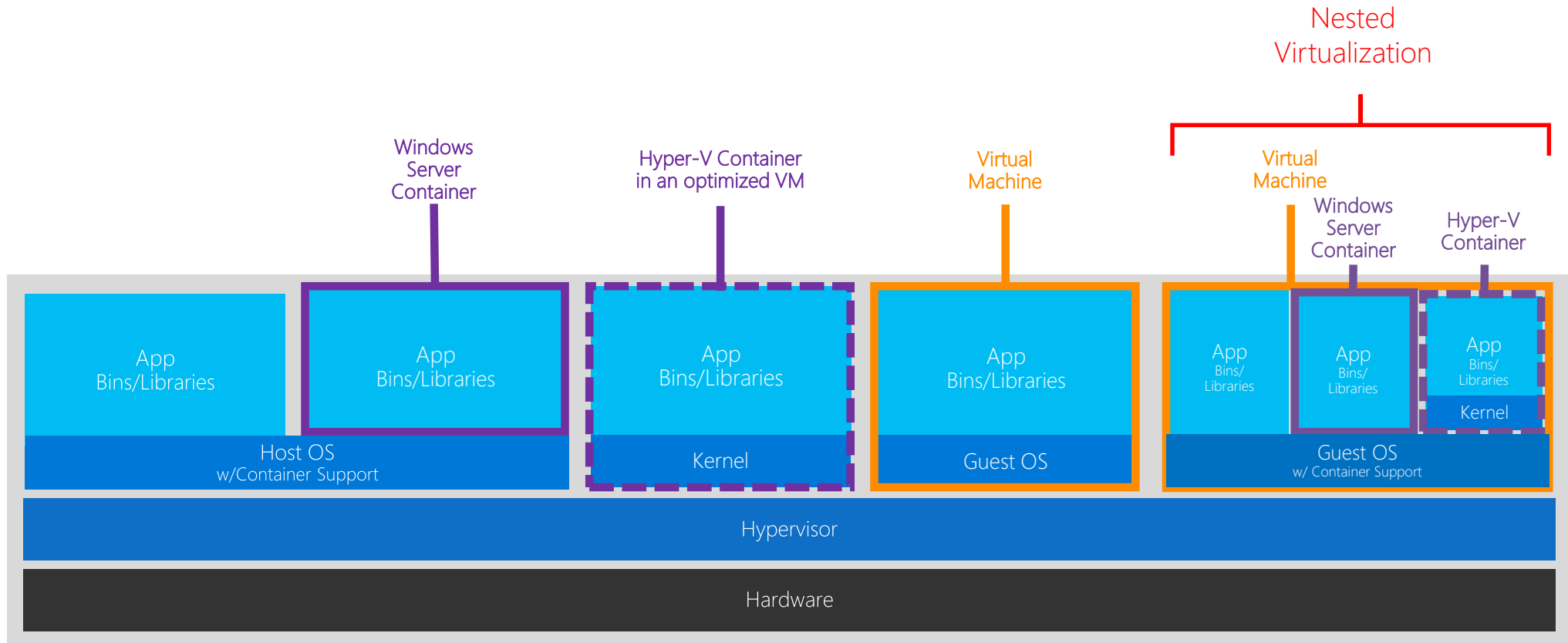
Isolation plus performance



Which one and why?

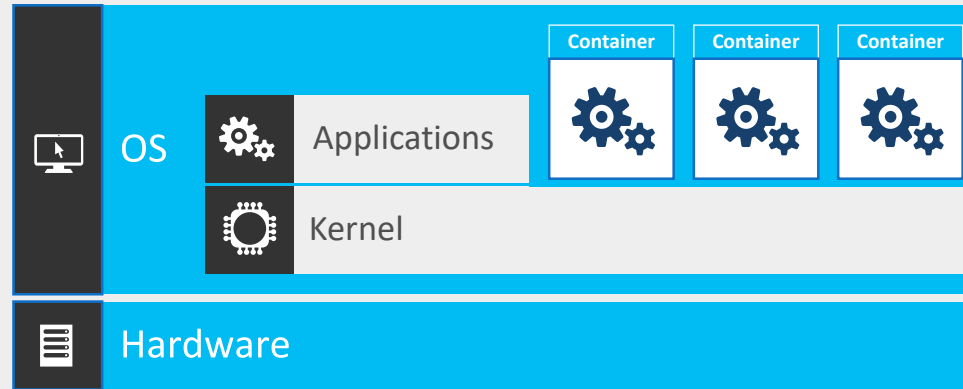
They are deployment and not design decisions.

VMs and the two types of Windows containers

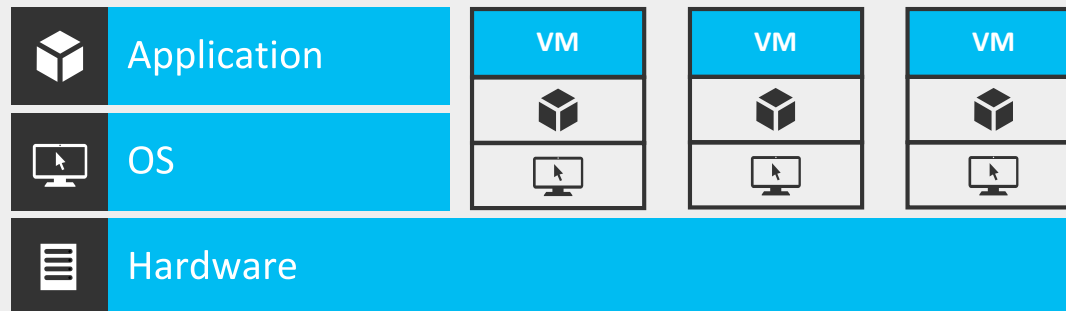


VMs and containers, friends or foes?

Containers = Operating system virtualization



Traditional virtual machines = hardware virtualization



- OS
- Size
- Speed
- Resource consumption
- Isolation
- Security
- Compliance
- Application requirements

Call to action

- Review your business requirements
 - Isolation level
 - Security and compliance
 - Size and speed
- Set your short-term and long-term goals
 - On-premises, cloud or hybrid
 - Lift and shift, rewrite
 - DevOps and CI/CD tools (Jenkins, Visual Studio Team Services, etc.)
- Evaluate VMs and containers
 - Linux, Windows
 - Orchestration platform (DC/OS, Kubernetes, Swarm, etc.)

Thank you



<http://yungchou.wordpress.com>
@yungchou