

FUTURE DECODED

6-7 OTT '16 / MILANO

IN PARTNERSHIP WITH:



CommunityDays.it

www.futuredecoded.it

 #FutureDecoded

Docker e il supporto ai Container in Azure e Windows Server

Igor Pagliai
Microsoft Principal Program Manager
DX TED Commercial Team (Corp.)

www.futuredecoded.it

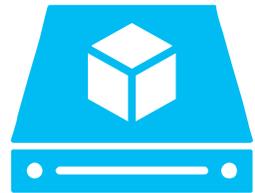
 #FutureDecoded

Agenda

- Introduction
 - Why «Containers»
 - What is «Containers»
 - Docker and Windows Containers
- First touch on main Docker API
 - Build Images & Run Containers
 - Networking & Storage
- Docker & Microsoft partnership
 - Tools & Technologies
 - Windows Server 2016 & Azure ACS
- Orchestrators
 - Cluster Managers

Evolution to Containers

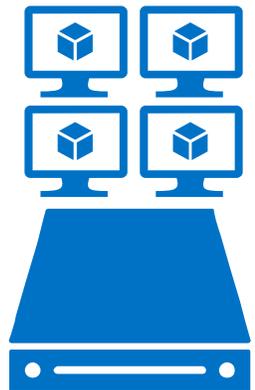
A new approach to build, ship, deploy, and instantiate applications



Physical

Applications traditionally built and deployed onto physical systems with 1:1 relationship

New applications often required new physical systems for isolation of resources



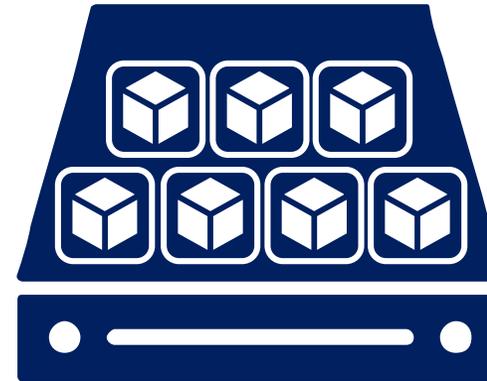
Virtual

Higher consolidation ratios and better utilization

Faster app deployment than in a traditional, physical environment

Apps deployed into VMs with high compatibility success

Apps benefited from key VM features i.e. Live migration, HA



Physical/Virtual

Package and run apps within
Containers

Key Benefits

Further accelerate of app deployment

New declarative image format

Reduce effort to deploy apps

Streamline development and testing

Lower costs associated with app deployment

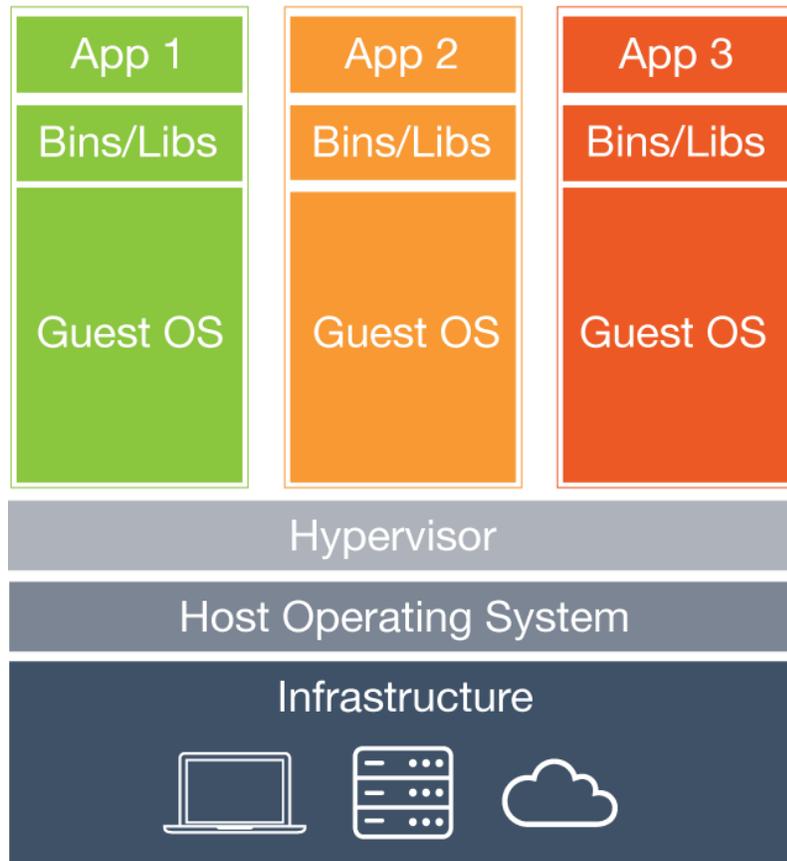
Increase server consolidation

Excellent for Micro-services

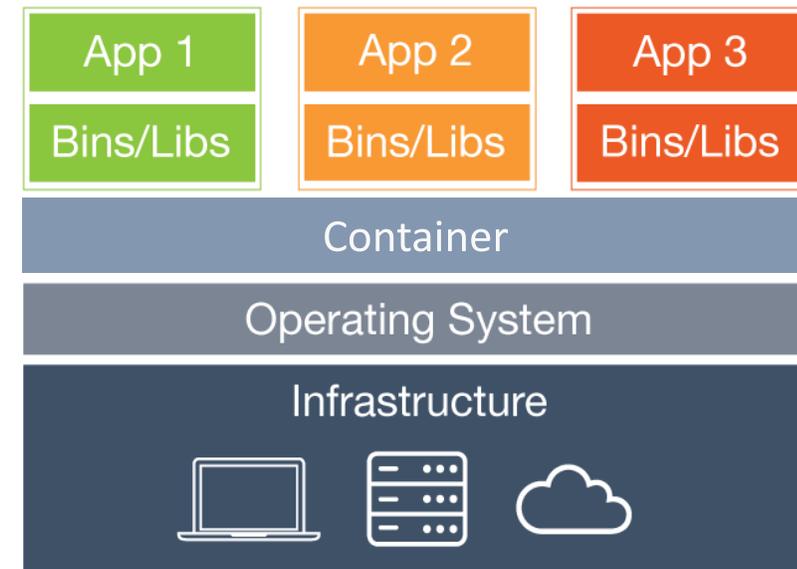
Faster scale & “on-line” upgrades

Containers are significantly more lightweight than a VM

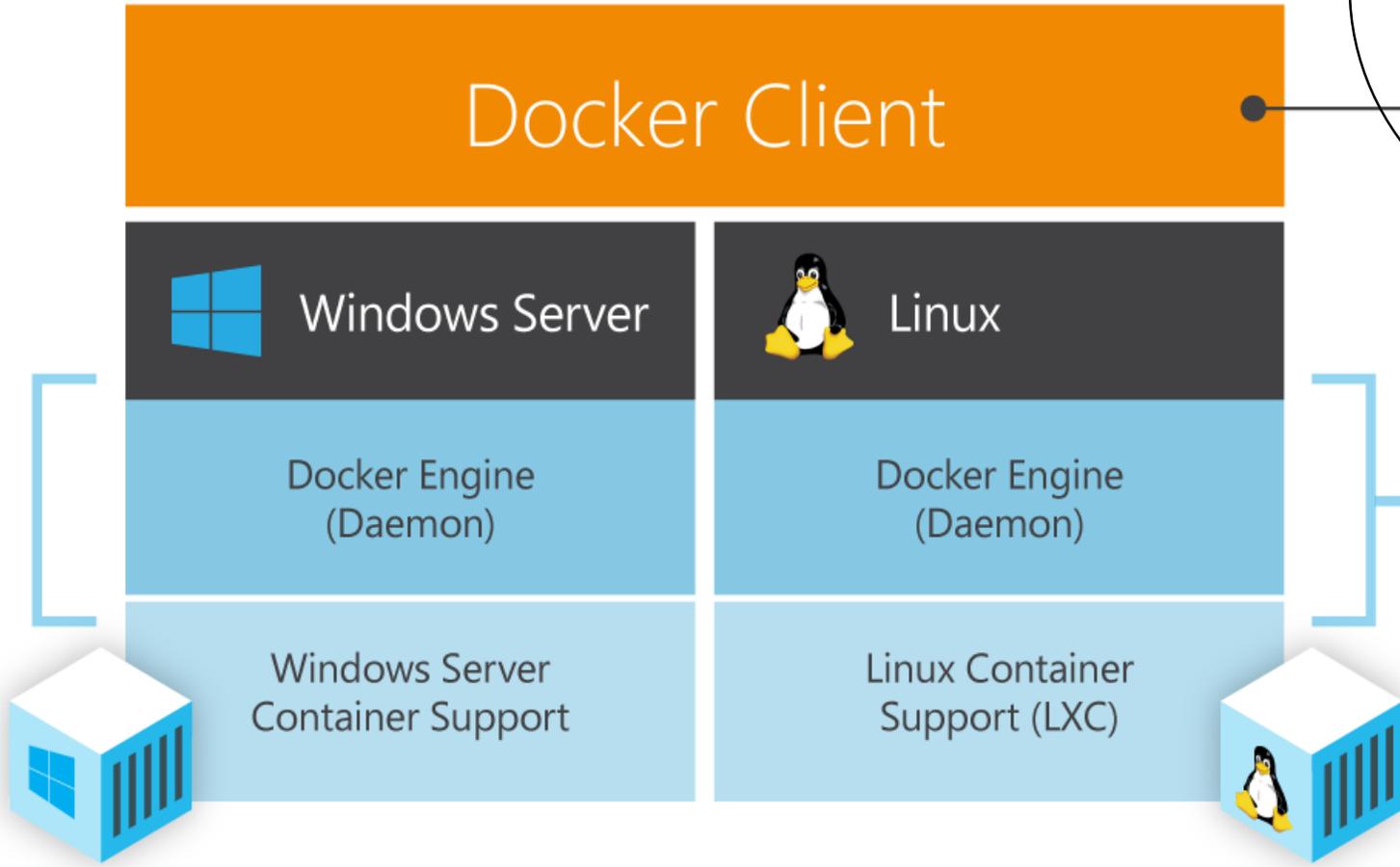
Virtual Machine



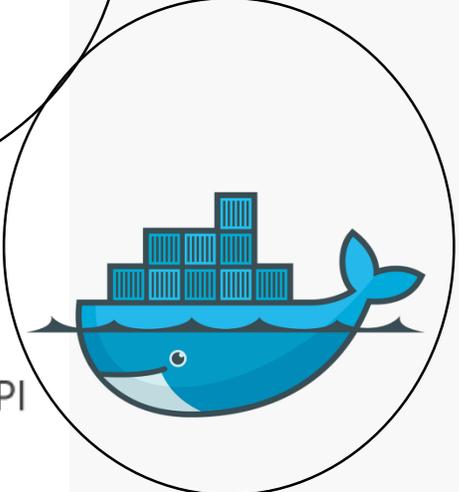
Container



Docker vs Windows Containers



Docker.exe
Examples:
docker run
docker images



Docker Remote API
Examples:
GET /images/json
POST /containers/create

Docker, Docker, Docker

Docker Inc. did not invent them

Container != Docker != Docker Inc

Docker is a container format and a set of tools

Docker CLI, Docker Engine, Docker Swarm, Docker Compose, Docker Machine, Registry and more...

Abstraction of where services are running

- Abstraction of where services are running
- Optimized for fast deployment, instancing, updating
- Services have complete isolation

What does Docker Inc. do?

- Docker Engine - open source container management.
- Docker Hub - online home and hub for managing your Docker containers.
- Docker Enterprise Support - commercial support for Docker.
- Docker Services & Training - professional services and training to help you get the best out of Docker.

Focused on Docker and growing the Docker ecosystem:

- Founded in 2009.
- Formerly dotCloud Inc.
- Released Docker in 2013.

Docker Architecture

- **Docker Daemon**

- Processes Docker API requests

- **Docker Client**

- Docker binary cmd line tool implements the Docker API and communicates with the daemon

- **Docker Hub Registry**

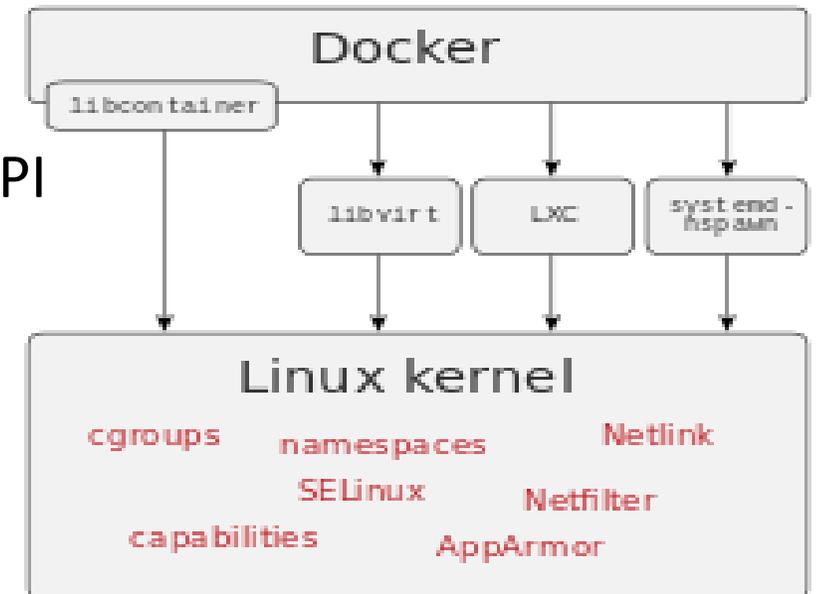
- Public repository for the docker images
- Docker daemon communicates with it via the Docker API

- **Cluster Managers / Orchestrators**

- Swarm, Mesos, Kubernetes, etc.

- **DOCKERFILE**

- “de-facto” standard for “Container” images



Docker differentiators

- Other Containers has proprietary formats
 - Docker supports portable deployment across different h/w, environments (QA → Prod)
- Versioning
 - Docker includes GIT-like capabilities for tracking container versions, inspecting the diff, etc.
- Docker images are comprised of a series of layers
 - This has a number of implications; one of which is that images can be built on top of other images
 - Docker Image is a read-only file system. Docker Container is a set of processes running in a read-write copy of the file system
- Shared libraries
 - There is already a public registry (<http://index.docker.io/>) where thousands have already uploaded the useful containers they have created => huge community

Demo

```
PS docker search microsoft
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
microsoft/dotnet	Preview images for the .NET Core command l...	107		[OK]
microsoft/azure-cli	Docker image for Microsoft Azure Command L...	52		[OK]
microsoft/iis	Internet Information Services (IIS) instal...	11		
microsoft/oms	Monitor your containers using the operatio...	3		[OK]
microsoft/applicationinsights	Application Insights for Docker helps you ...	3		[OK]
microsoft/sample-dotnet	.NET Core running in a Nano Server container	1		
microsoft/dotnet35		1		

```
PS docker pull microsoft/dotnet
```

```
Using default tag: latest
```

```
latest: Pulling from microsoft/aspnet
```

```
604d05dfd165: Extracting [=====>] 34.78 MB/37.19 MB
```

```
a3ed95caeb02: Download complete
```

```
af271166b5e5: Download complete
```

```
ffff72610562: Pull complete
```

```
82633c2ea8fc: Pull complete
```

```
1ec63dc1715b: Downloading [=====>] 29.72 MB/64.67 MB
```

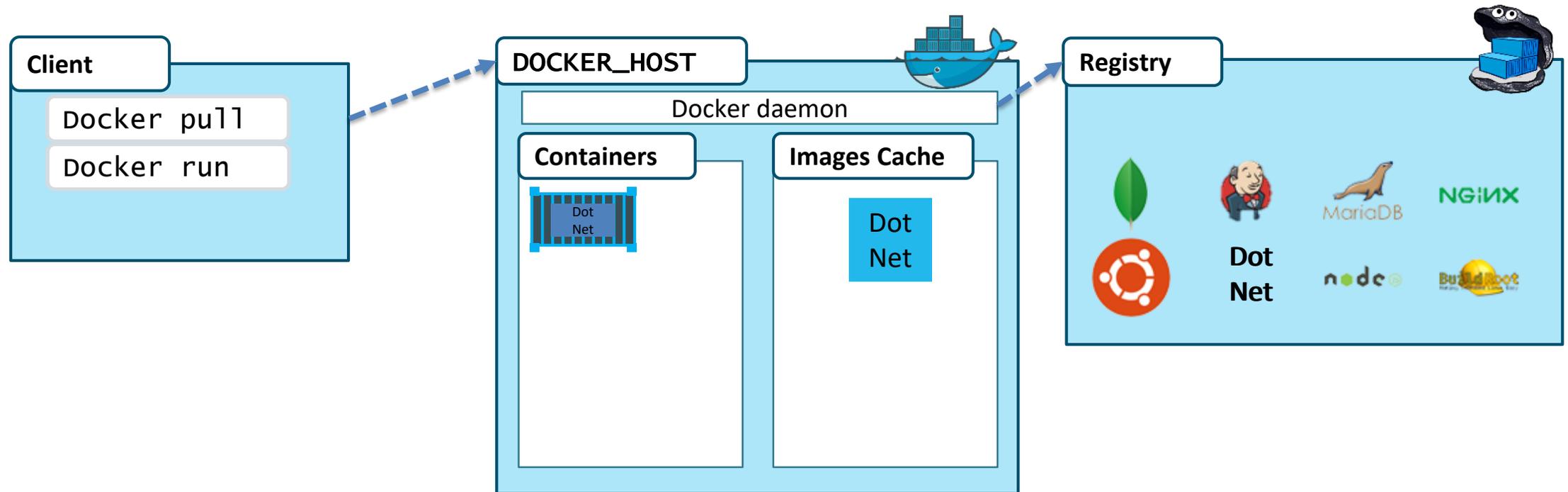
```
9be54d5ae146: Pull complete
```

```
c3abf6246d6a: Pull complete
```

```
Digest: sha256:cbbc0a7be0926d017ffd952799c4864c2c6cf812cbb70862bf2c8539401aa1e3
```

```
Status: Downloaded newer image for microsoft/dotnet:latest
```

Docker API: A first touch !

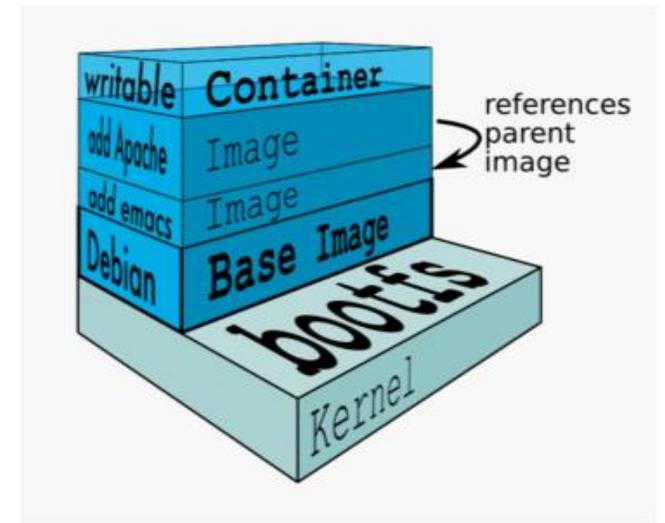


Anatomy of a Docker image

An “image:tag” points to a “layer ID” (.tar):

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
ubuntu	15.04	c8be1ac8145a	6 weeks ago



A docker image consists of layers:

```
$ docker history ubuntu:15.04
```

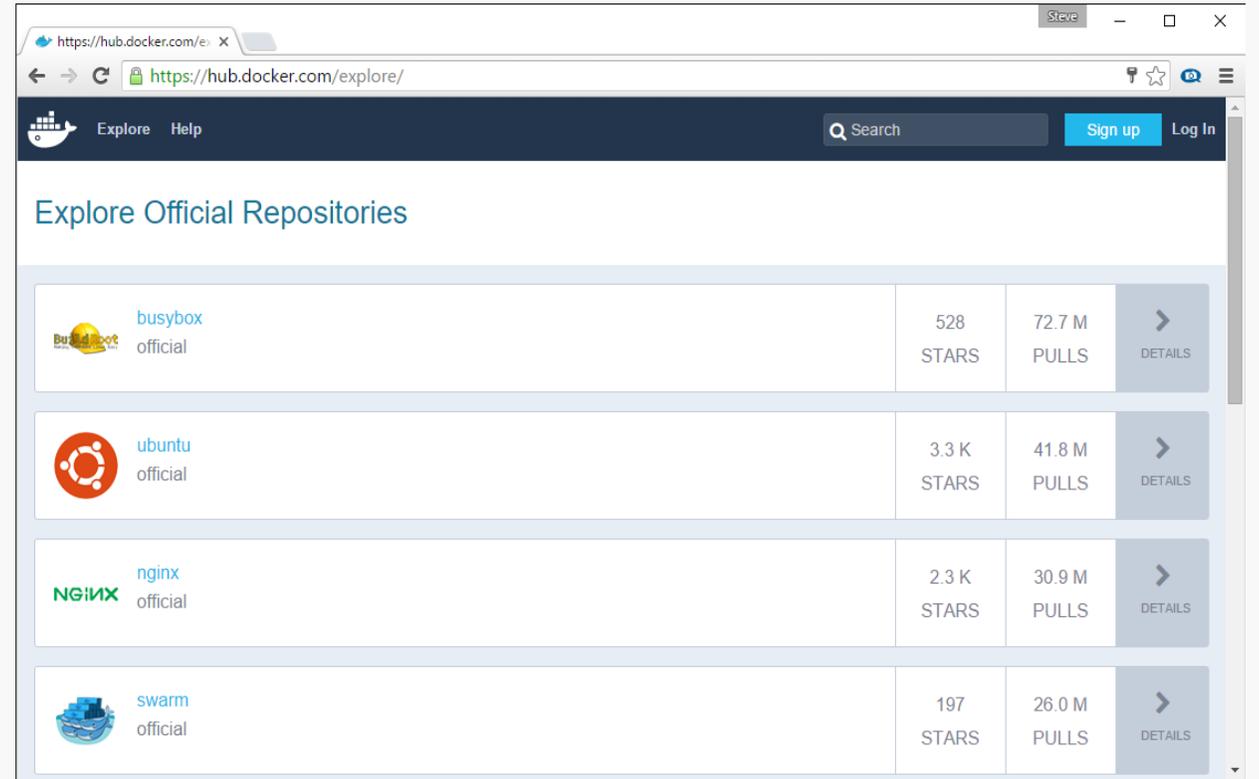
IMAGE	CREATED	CREATED BY	SIZE
c8be1ac8145a	6 weeks ago	/bin/sh -c #(nop) CMD ["/bin/bash"]	0 B
e65155041eed	6 weeks ago	/bin/sh -c sed -i 's/^#\s*\s*(deb.*universe\)\$/'	1.879 kB
df6e891a3ea9	6 weeks ago	/bin/sh -c echo '#!/bin/sh' > /usr/sbin/polic	701 B
47984b517ca9	6 weeks ago	/bin/sh -c #(nop) ADD file:8e4943cd86e9b2ca13	131.3 MB

All layers (except the topmost layer) has a parent layer.

[Image2Docker](#): Converting VHD/VHDX/VMDK/WIM to DOCKERFILE files

Docker Registry

- Stores docker images
- Searchable
- Public Registry – [Hub.Docker.com](https://hub.docker.com)
Registry Server
- Private Registries –
Instanced for you
 - Can be hosted in Docker, Azure, AWS, Google, ...



Demo - build & push

```
PS docker login
```

```
Username (stevelasker):
```

```
Password:
```

```
WARNING: login credentials saved in C:\Users\SteveLas\.docker\config.json
```

```
Login Succeeded
```

```
PS docker tag hello1d stevelasker/hello1d:v1
```

```
The push refers to a repository [docker.io/stevelasker/hello1d:v1]
```

```
5f70bf18a086: Pushed
```

```
30782ae9410d: Mounted from microsoft/dotnet
```

```
ca3a6ab52687: Pushed
```

```
162da15b5454: Pushing [=====>] 103.5 MB/122.6 MB
```

```
162da15b5454: Pushing [=====>] 9.201 MB/44.41 MB
```

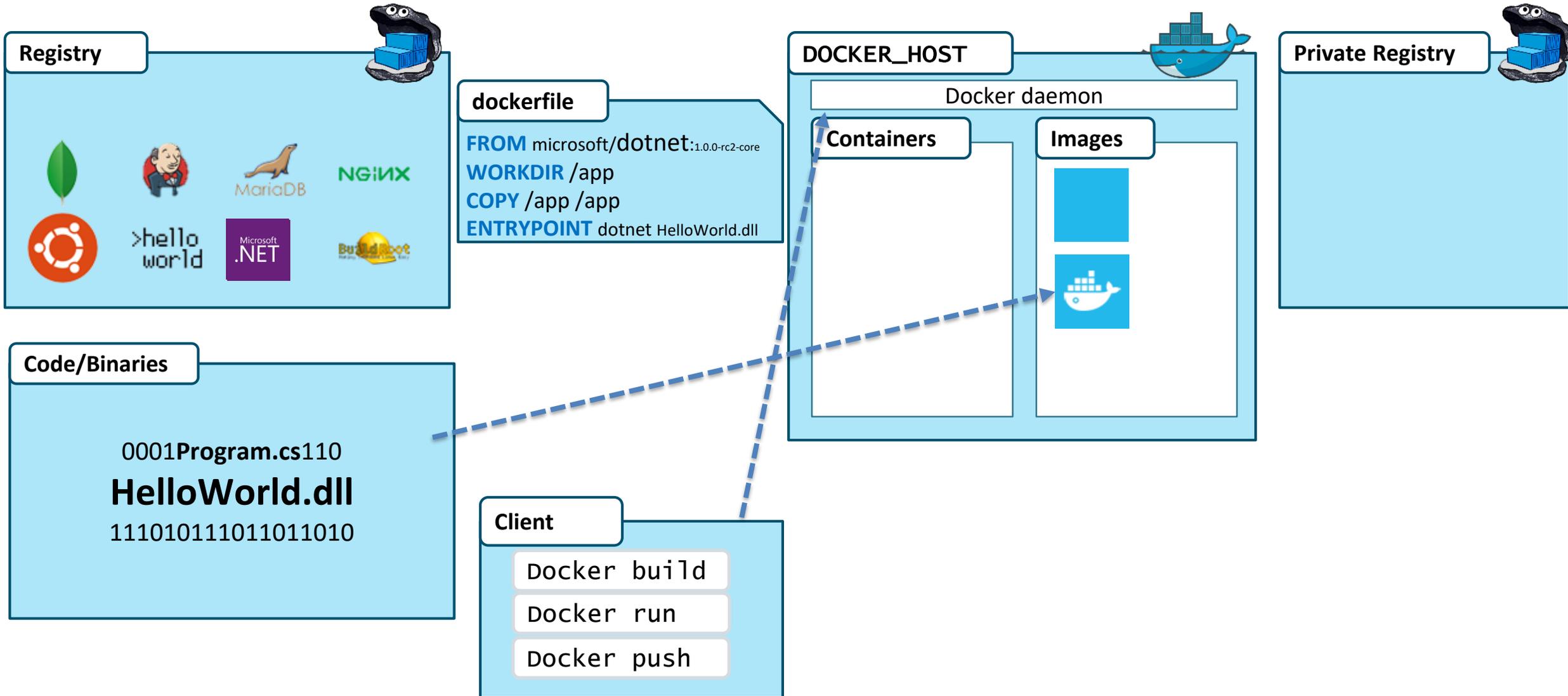
```
ddef881aeaff: Waiting
```

```
v1: digest: sha256:71acc144df8563d7f319e5c65108490501a8a67364db5517db792d721d92dfc0 size: 6303
```

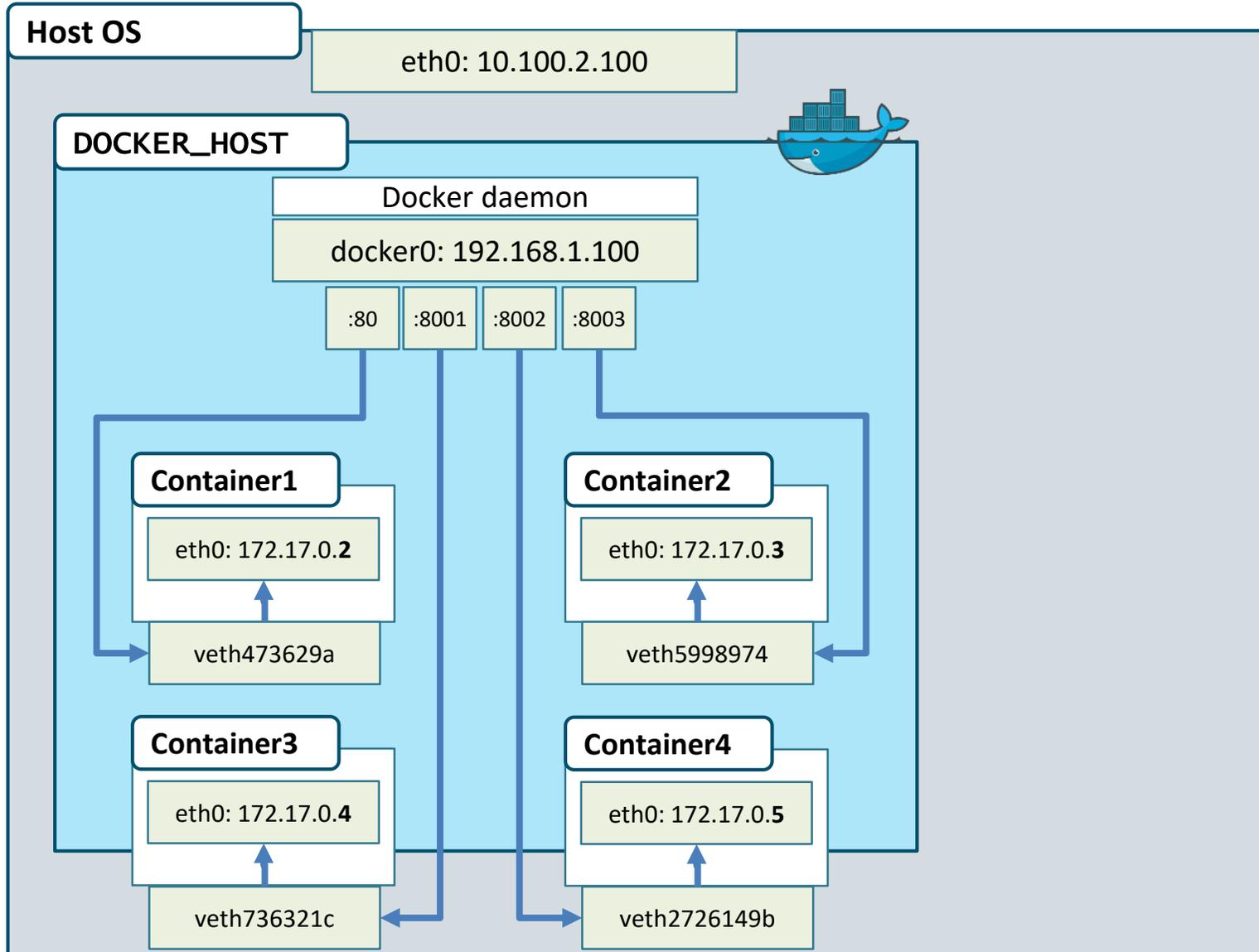
```
# The following can now be run from any client
```

```
PS docker run stevelasker/hello1d:v1
```

Docker build



Networking - NAT



Docker Networking uses NAT to route inbound traffic to each container. Each container can use the same port, internal to its container, but must be addressable outside by unique ports per HOST.

Load Balancers and Service Discovery are new important topics.

Note: networking may vary based on your orchestration engine

Container Networking

- A NAT network created by default when the DOCKERD service first runs
 - The default internal IP prefix created is 172.16.0.0/12
 - Container endpoints will be attached to this default network and be assigned an IP address from the internal prefix
- Windows containers support four different networking drivers or modes
 - NAT, Transparent, L2bridge, and L2tunnel
 - On Windows, managed using [PowerShell for Docker](#) & `DOCKER NETWORK CREATE/LS`
 - *Run a container and attach to the*
`docker run -it --network=MyNatNetwork`
- Port Mapping
 - Static: `docker run -it -p 8080:80`
 - Dynamic: `docker run -it -P` (rely on DOCKERFILE “EXPOSE” clause)
 - Randomly chosen ephemeral allocated, check with DOCKER PS command

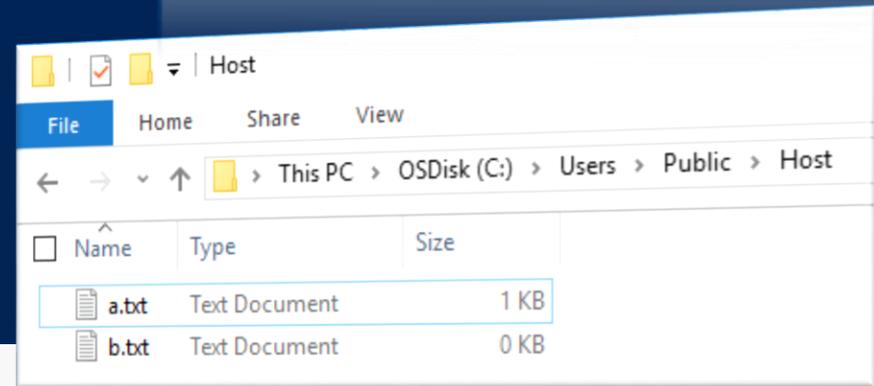
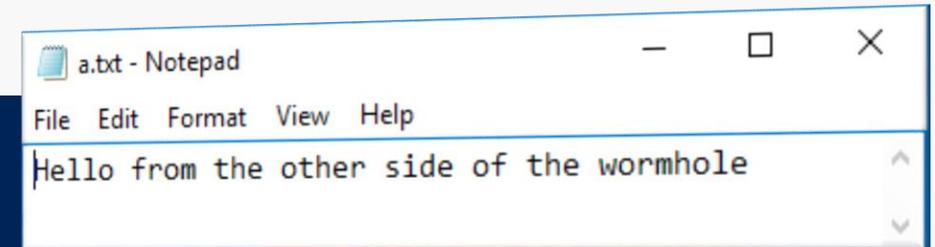
DEMO: Docker Volume & Persistent Storage

Docker Containers are Immutable

- To save data, you must provide external storage
- Volume Drivers: Azure Files, SMB Shares, [Flocker](#) & OSS, On-prem vendors

Docker RUN **-v** hostpath:/containerpath

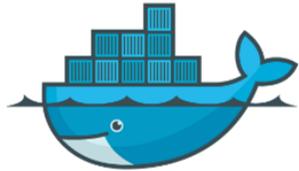
```
PS docker run -it -v /c/Users/SteveLas:/wormhole busybox
/ # ls
bin      dev      etc      home     proc     root     sys
/ # cd wormhole
/wormhole # touch a.txt
/wormhole # touch b.txt
/wormhole # ls
a.txt  b.txt
/wormhole # cat a.txt
Hello from the other side of the wormhole
/wormhole #
```



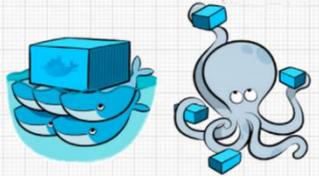
Docker **CP** [OPTIONS] CONTAINER:SRC_PATH DEST_PATH

Docker & Microsoft Partnership

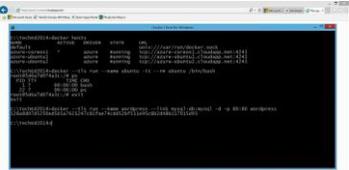
Docker extensions
in Microsoft Azure



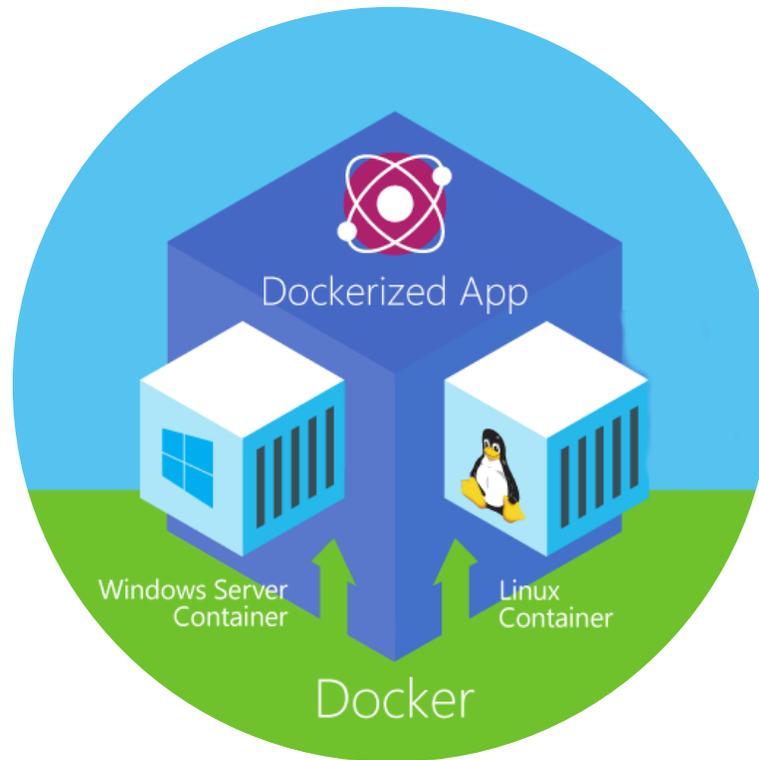
Docker Daemon
Docker-Compose
Docker-Machine
Docker Swarm



Docker client
for Windows



Docker VM
image in Azure



Docker | Microsoft

ASP.NET & Core .NET
Docker images

SQL Server is coming
(with many others...)

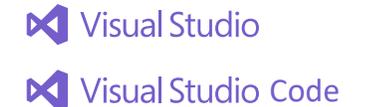


ASP.NET 5 Preview Docker Image

Azure ACS
Swarm & Mesos



Developer
tools
(VS, VSTS, Jenkins,
PowerShell, etc.)



Windows Server
Containers showcase



Windows Server

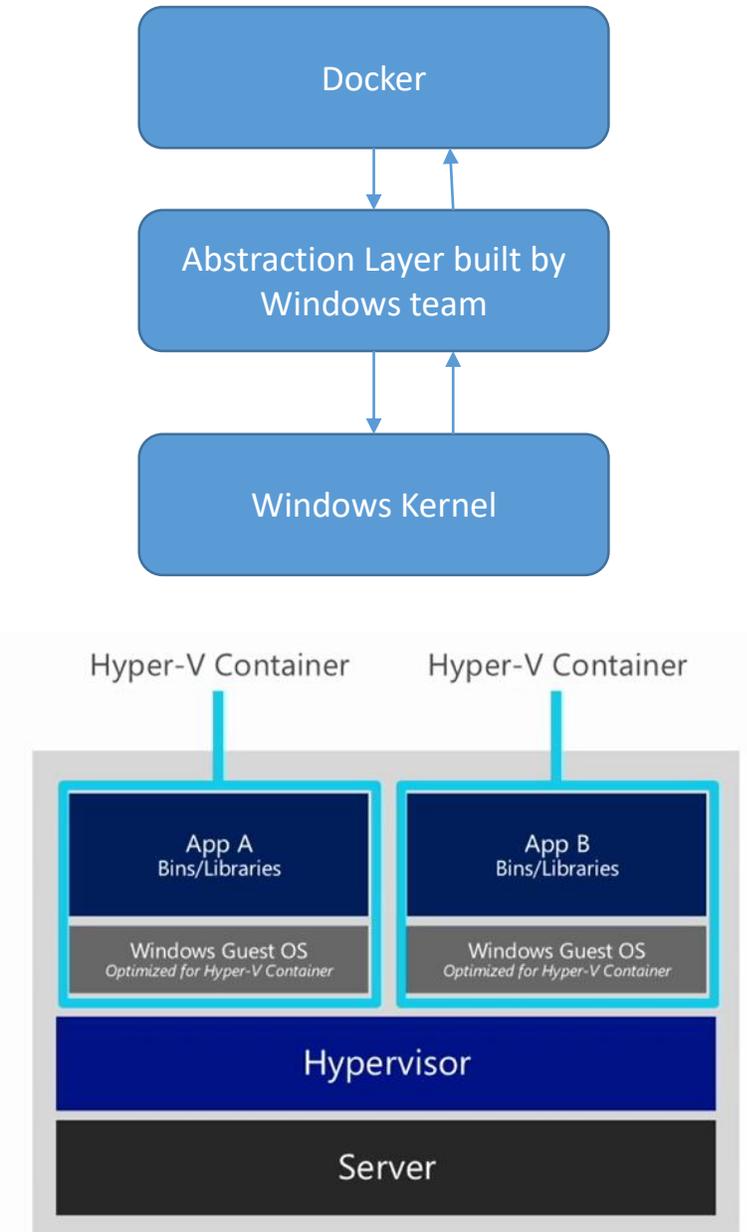
INNOVATION

Docker and Windows Containers

- Docker being ported to work on Windows Server 2016
 - No additional virtualization layers
 - Container runs Windows, not Linux
- Container Types
 - Windows (Server) Containers
 - Hyper-V Containers
- Core system arch work done on Windows end
 - Job scheduler extended to support “Namespaces”
 - NTFS semantics overlay over Docker Directory
 - Try it off Azure portal, TP5 available in Azure Gallery

Hyper-V containers look similar but don't share the OS, provide another layer of isolation →

docker run **--isolation=hyperv** microsoft/nanoserver



Windows Server Containers

Anatomy and key capabilities



Spotlight capabilities

Build: Developers will use familiar development tools, such as Visual Studio, to write apps to run within containers.

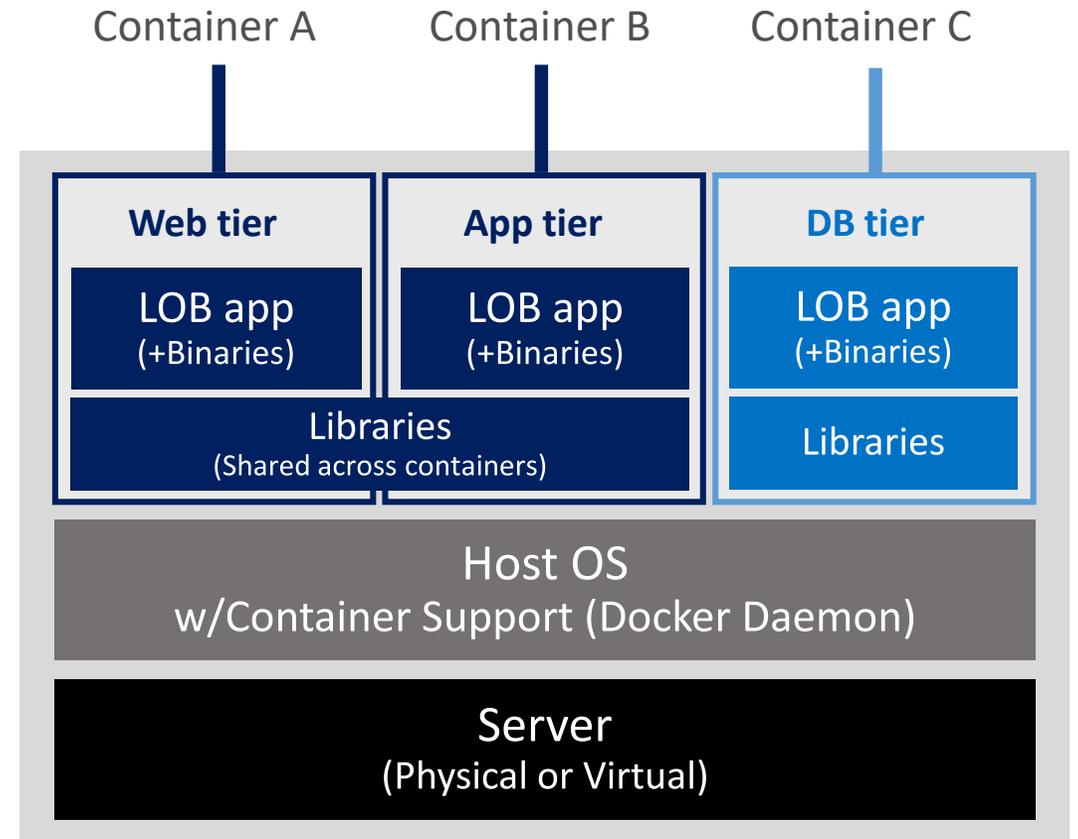
By building modular apps leveraging containers, modules can scale independently, and be updated on independent cadences.

Run: Container capabilities built into Windows Server

Manage: Deploy and manage containers using PowerShell, or using Docker.

Resources: Define CPU and memory resources per container along with storage and network throughput.

Network: Provide NAT or DHCP/static IP for network connectivity.



Hyper-V Containers

Anatomy and key capabilities



Spotlight capabilities

Consistency: Hyper-V Containers use the same APIs Windows Server Containers ensuring consistency across management and deployment toolsets.

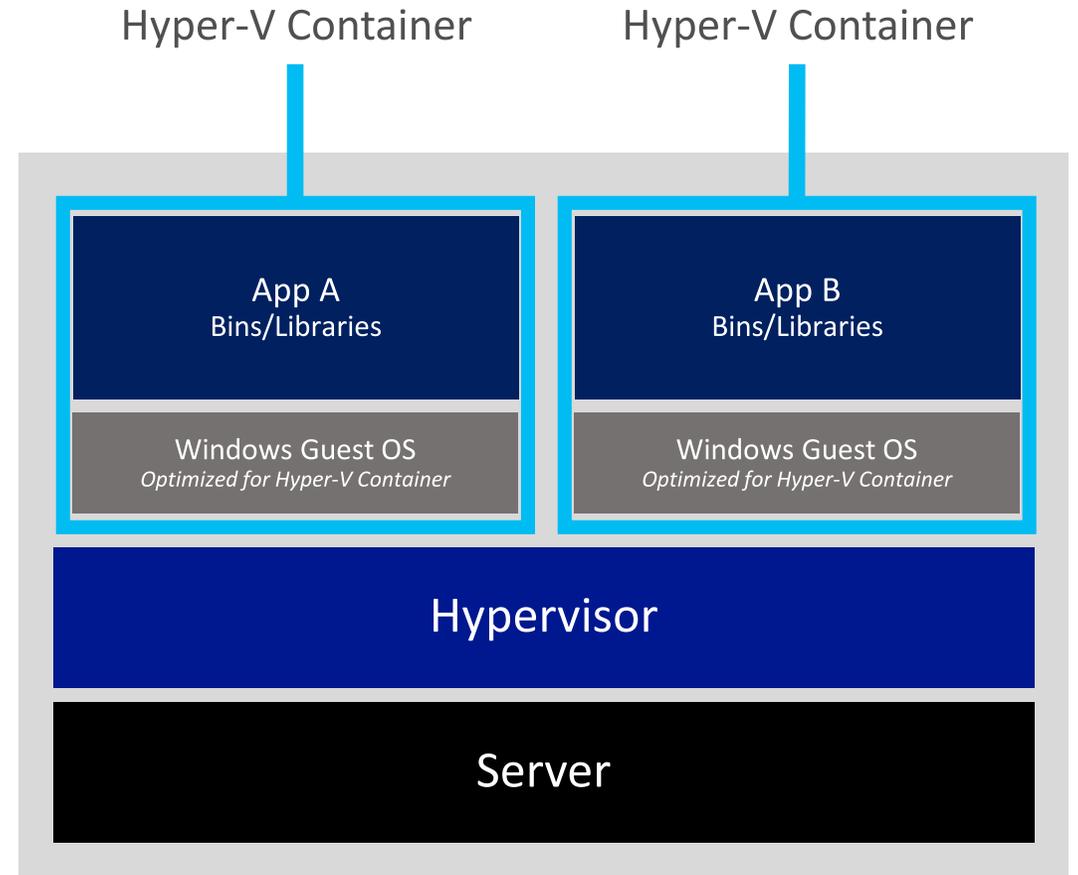
Compatibility: Hyper-V Containers use the exact same images as Windows Server Containers.

Strong Isolation: Each Hyper-V container has its own dedicated copy of the kernel.

Highly Trusted: Built with proven Hyper-V virtualization technology.

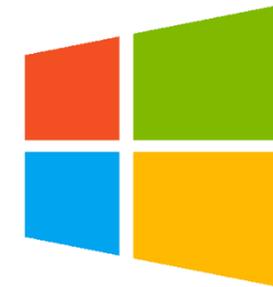
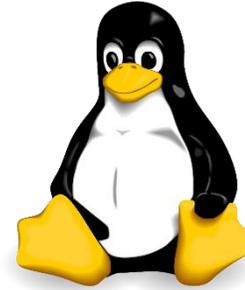
Optimized: The virtualization layer and the operating system have been specifically optimized for containers.

Hyper-V in Azure VM: Need “Nested Virtualization”, coming soon....



DEMO: What you have to play with....

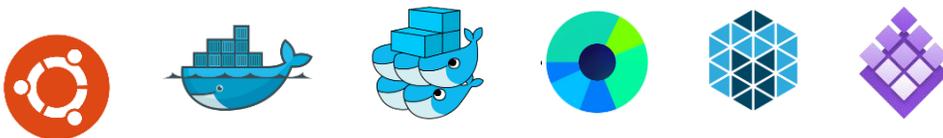
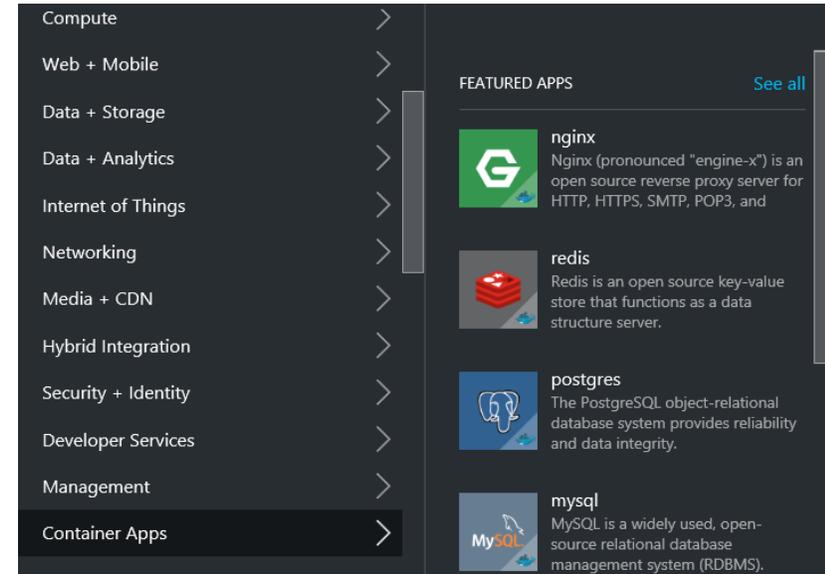
- **Azure CLI Docker Integration**
 - Azure client tools available on Windows, Mac and Linux
- **Docker Host images**
 - Ubuntu vs. Windows Server 2016
 - Azure Docker VM Extension
- **Docker Machine Azure Driver**
- **Docker for Windows vs. Docker on Windows**
 - Former from Docker Inc., latter from Microsoft Corp.
- **Docker Registry Azure Storage Driver**
- **Azure Container Service (ACS)**
 - For Linux & Windows (preview)
 - Orchestrators: Swarm, Mesos
- **ARM Templates**
 - Single VM deployments, ACS clusters
- **Azure File Service Docker Volume Driver**
 - Linux version, Windows coming



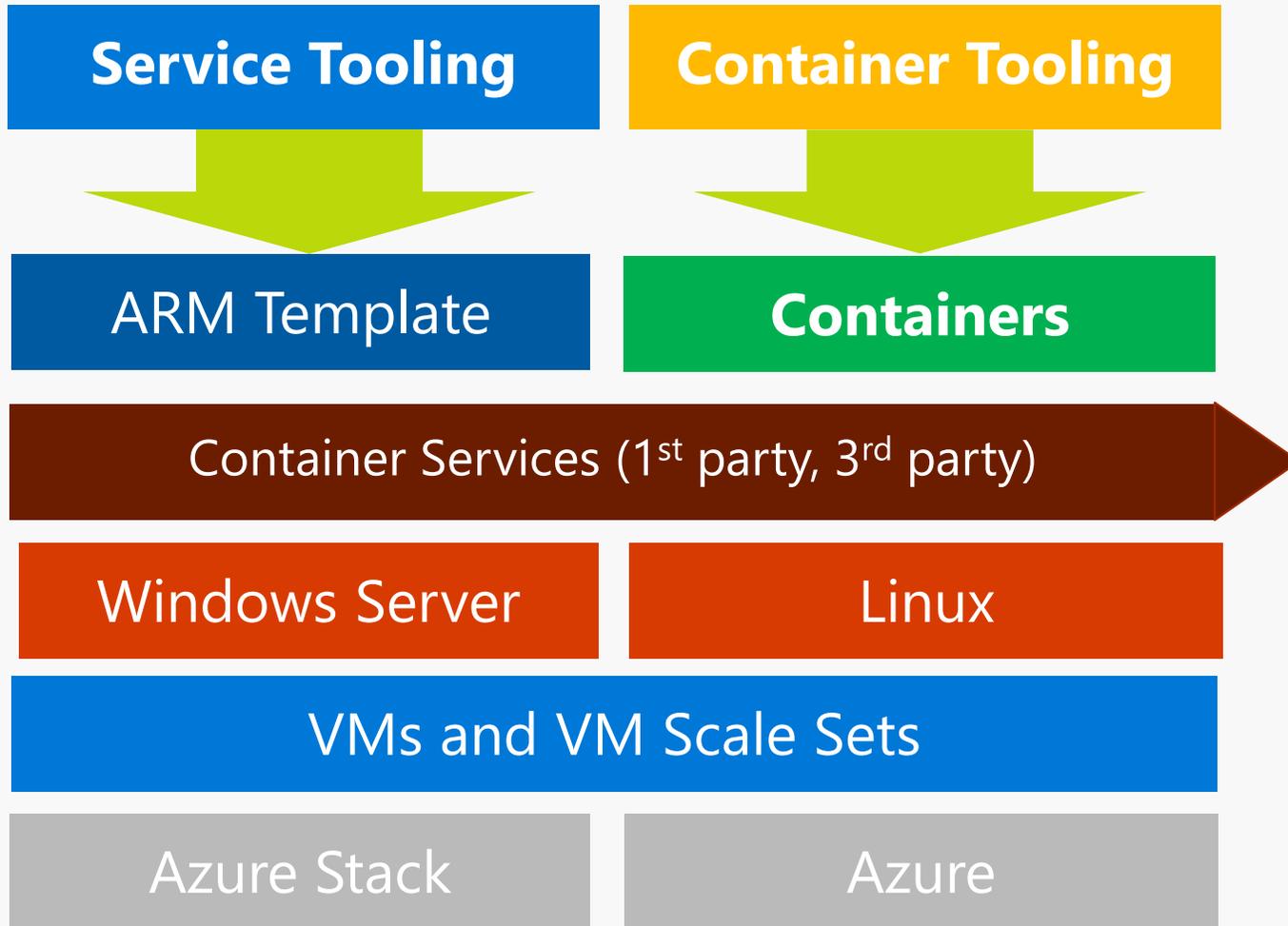
Docker and Azure Container Service

Container Apps

- **Azure Container Service** will make it easy for you to create and manage clusters of hosts pre-configured with Docker, Apache Mesos, Marathon and Docker Swarm
 - Simplifies the creation and configuration of a Mesos cluster setup with Docker
 - Docker tooling and API support
- **Container Apps** are pre-packaged docker images off marketplace
- **Support for Windows currently in preview**

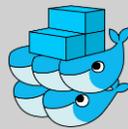


Azure Container Services

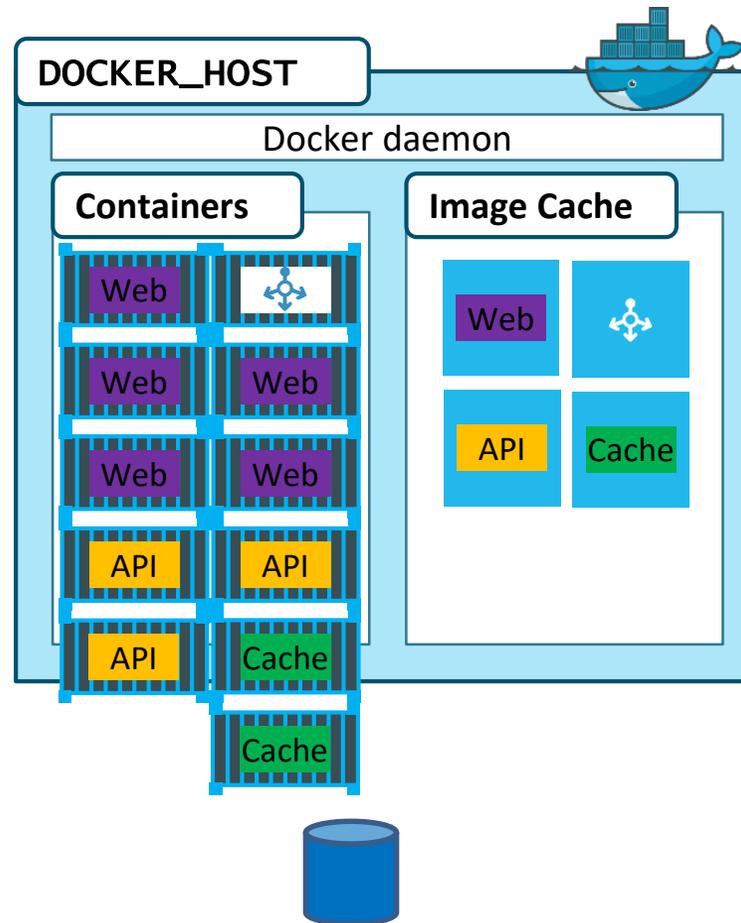


Layer	Supported Technologies
Configuration as Code	ARM, Dockerfile, Docker Compose, Marathon.json
Host cluster management	VM Scale Sets
Container orchestration	Docker Swarm, Chronos, Marathon, Apache Mesos
Monitoring	OMS, Statsd
Load Balancers	HAProxy, NGINX

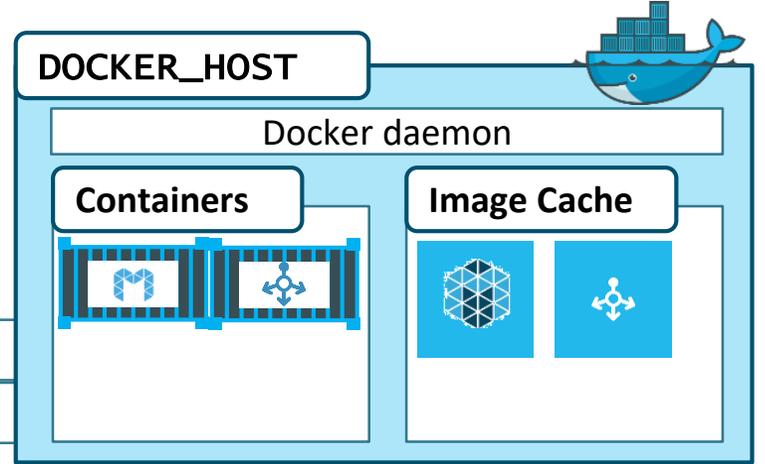
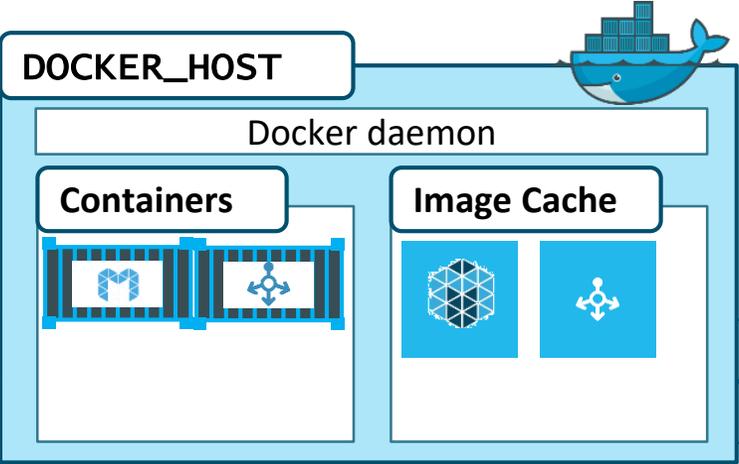
Azure Container Service flavours

	Docker Swarm	DC/OS (Mesos)
Container Dashboard	< NONE >	 DC/OS
Service Discovery	 Consul	 MESOS
Container Orchestration	 Swarm	 Marathon
Container Engine	 docker	 docker
Operating System	 ubuntu  Windows Server	 ubuntu

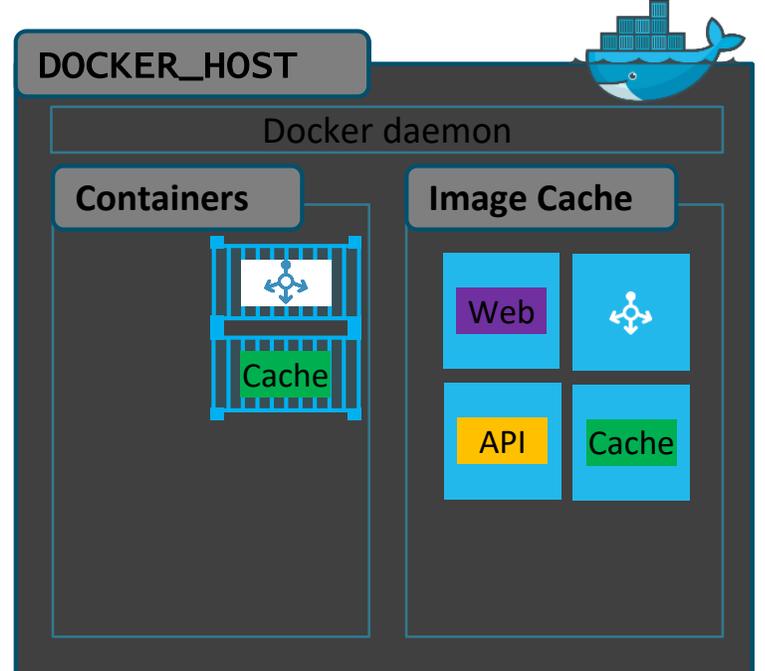
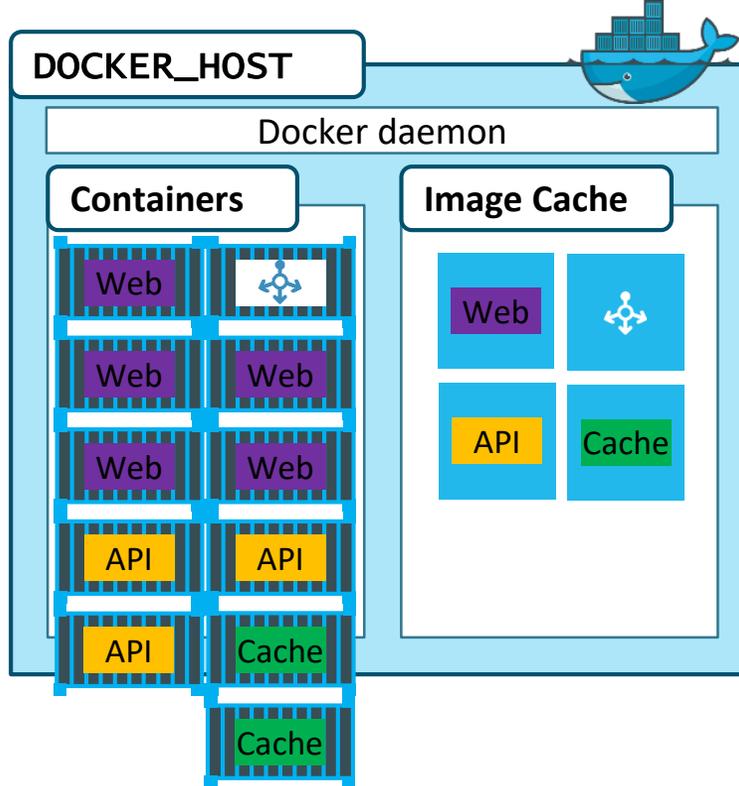
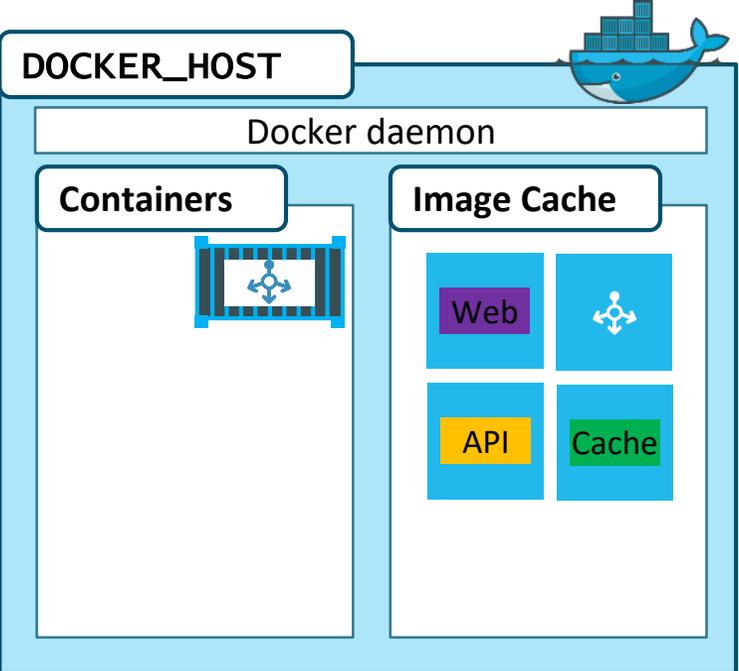
Load Balancing & Fault Tolerance



Distribution



Container Scheduling
Container Orchestration



Scaling, HA & LB: Orchestrator needed!

- OSS Mesos (& Marathon) -> Mesosphere DC/OS

- DCOS (Data Center Operating System) for node management, Marathon as app-hosting engine
- On Azure: <https://azure.microsoft.com/en-us/services/container-service/>
- Different ports/API, specific DCOS client
- Used by [Twitter](#), [Netflix](#), [Uber](#) and [NASA JPL](#)

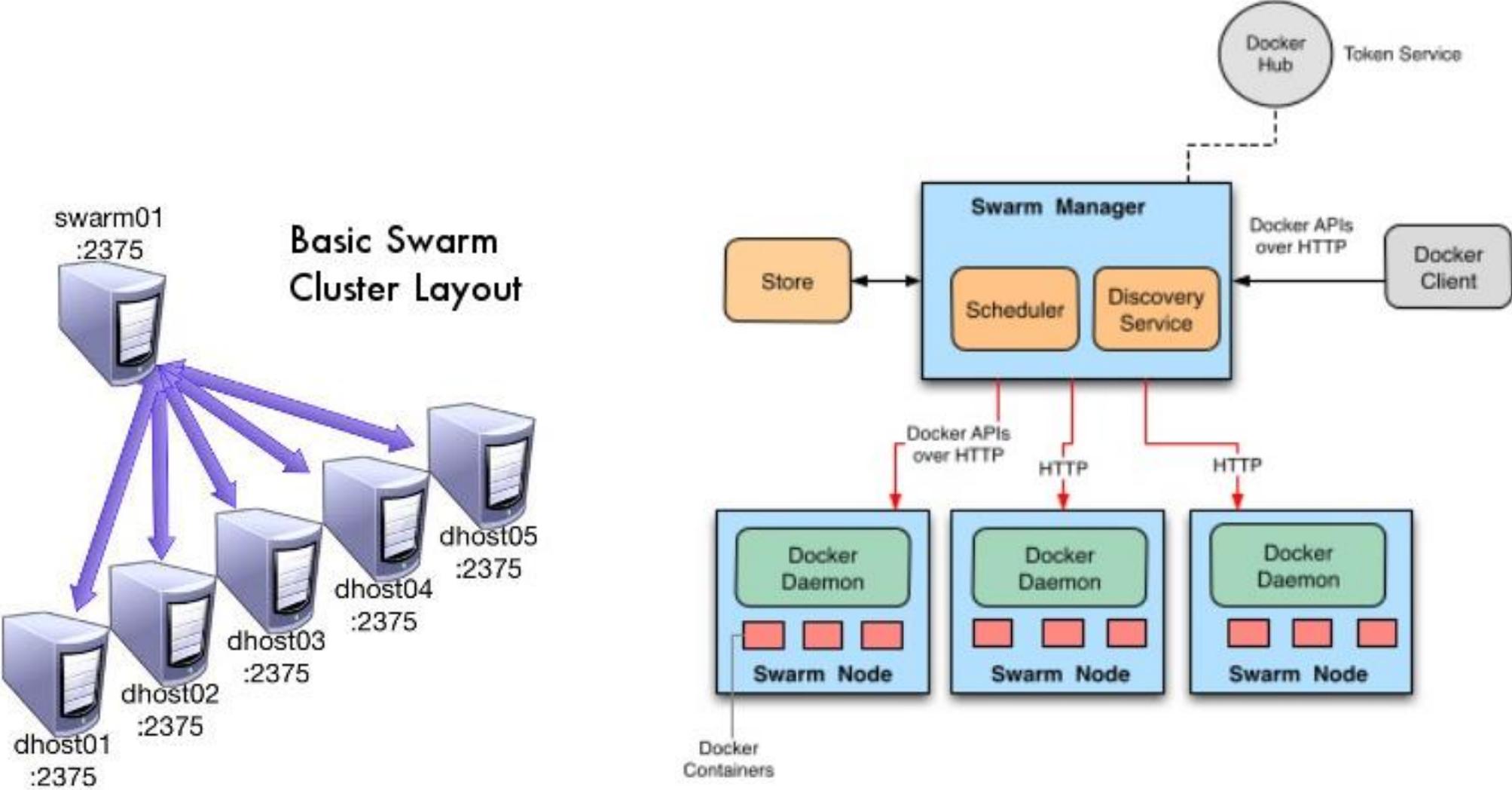
- OSS Swarm (& Consul) -> Docker Swarm

- Native orchestration engine built by Docker, full API compatible to Docker API
- SWARM mode in Docker V1.12 and later
- On Azure: <https://azure.microsoft.com/en-us/services/container-service>
- Also Azure: <https://azure.microsoft.com/en-us/documentation/articles/virtual-machines-docker-swarm/>

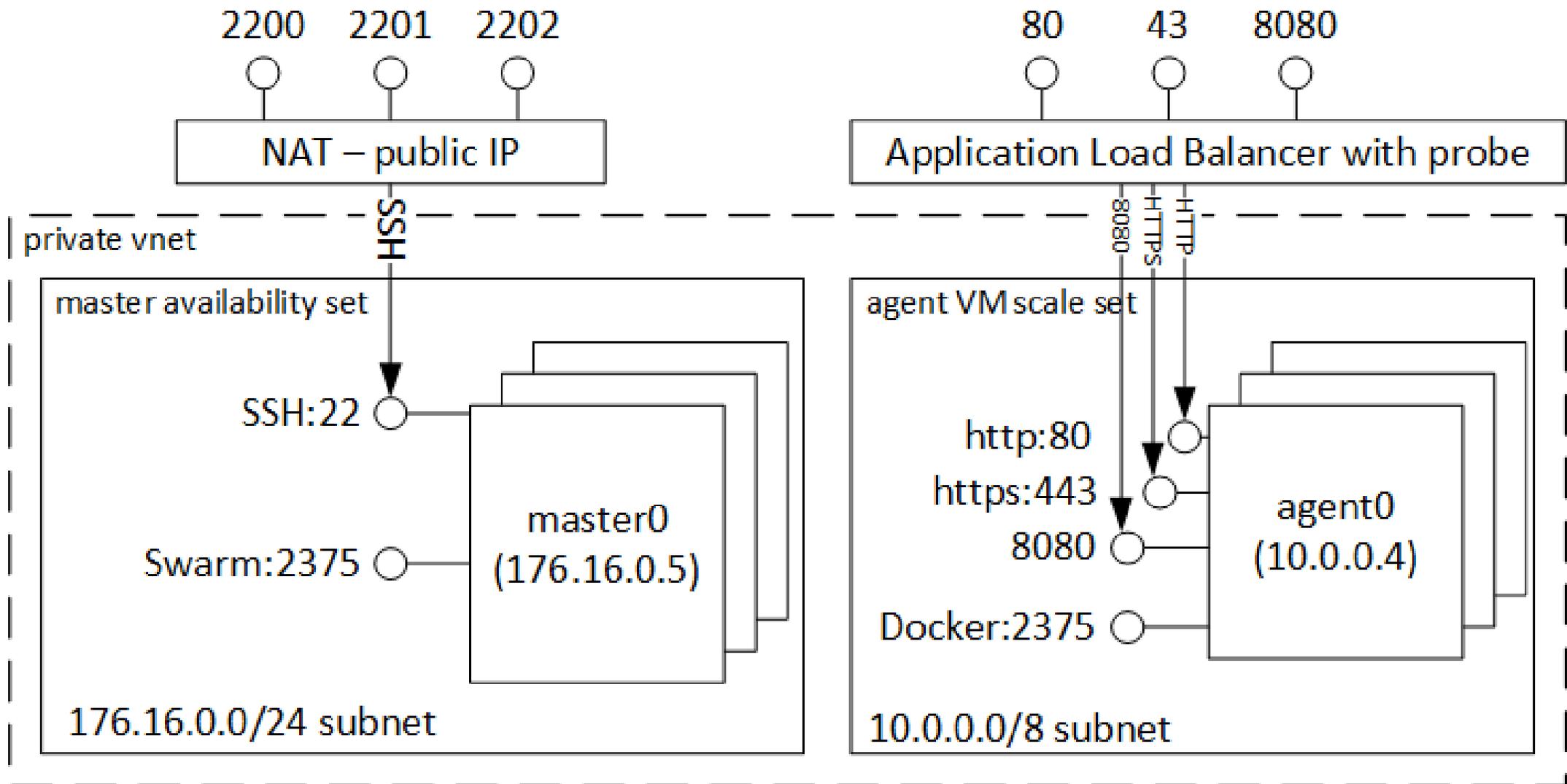
- Kubernetes

- Orchestration engine by Google; works with Docker; Google Container Engine as hosted version
- On Azure: <http://kubernetes.io/docs/getting-started-guides/coreos/azure/>

Docker Swarm

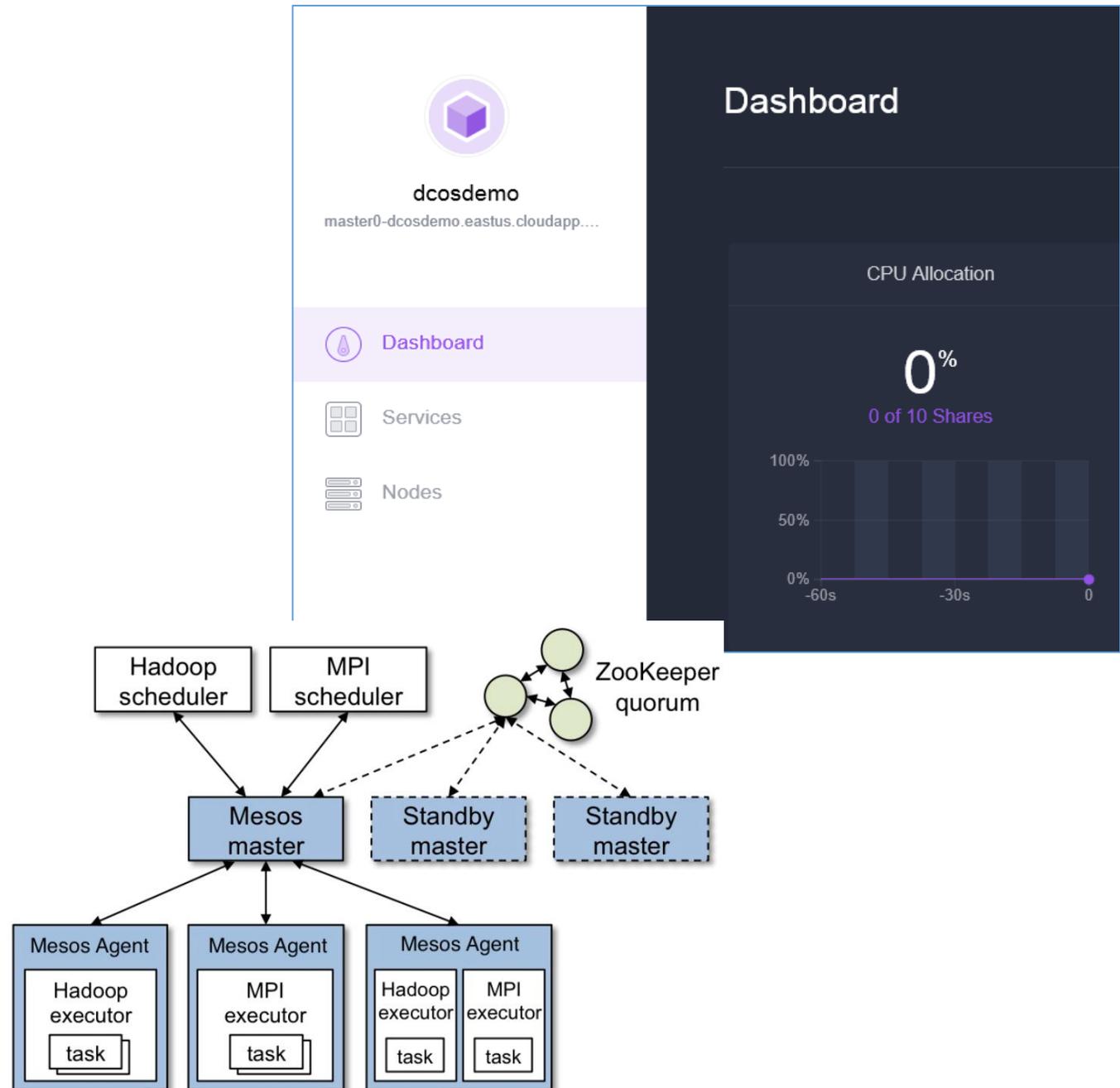


Azure Container Service Architecture (Swarm)

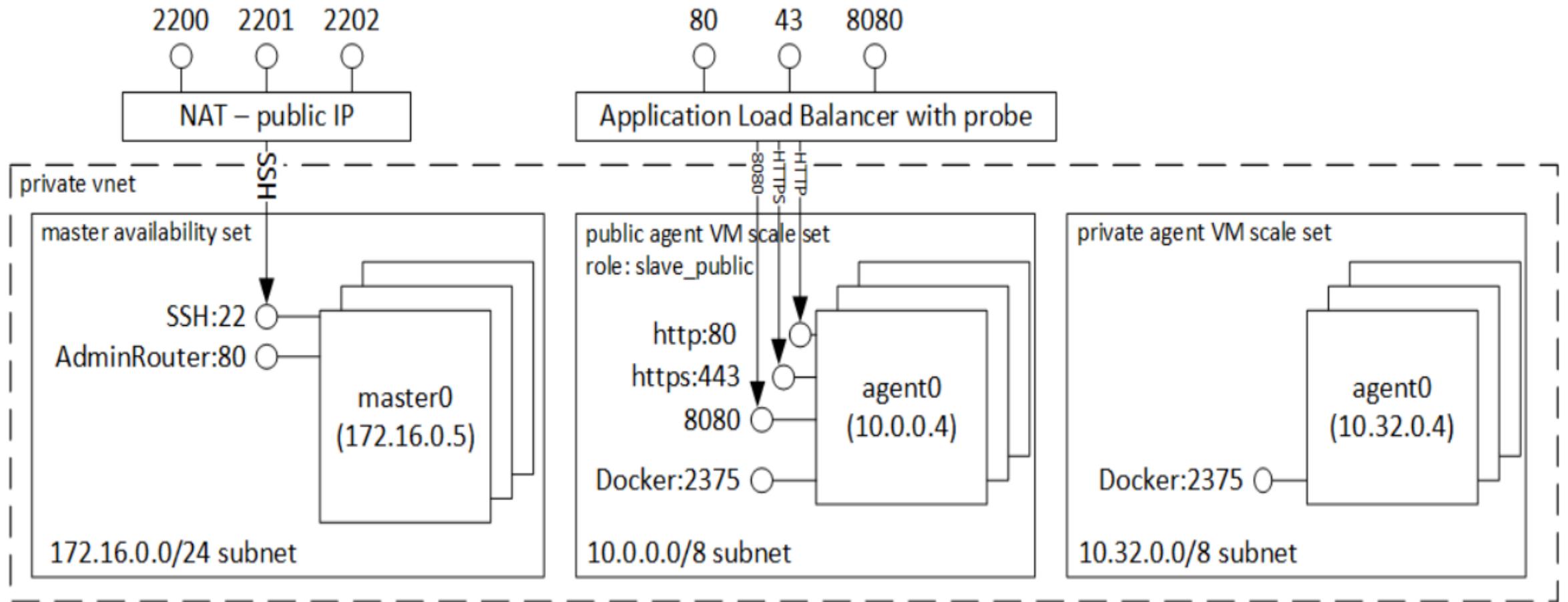


DC/OS in Azure

- Datacenter OS (DCOS) that is built on **Marathon** (scheduler) and **Mesos** (cluster manager)
- Scheduling, LB and Discovery
- Marathon LB using HAProxy
- Service Discovery
- Marathon LB vs. Mesos DNS
 - Azure uses the former
- OSS v1.0 project released on July 27th 2016

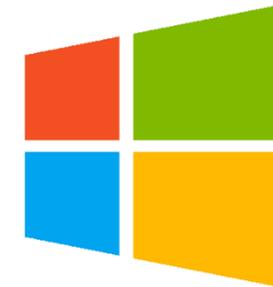
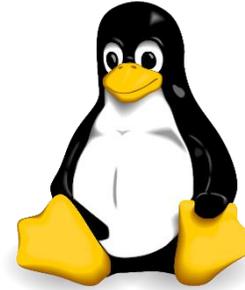


Azure Container Service Architecture (DC/OS)



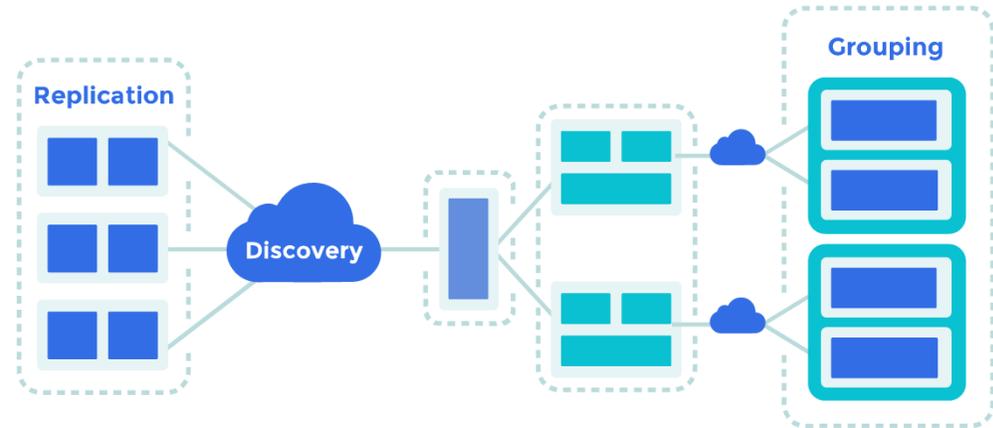
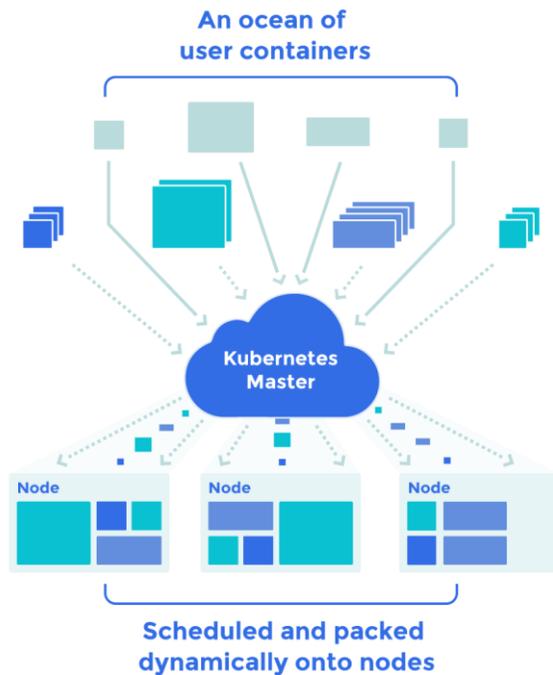
DEMO: What you have to play with....

- **Azure CLI Docker Integration**
 - Azure client tools available on Windows, Mac and Linux
- **Docker Host images**
 - Ubuntu vs. Windows Server 2016
 - Azure Docker VM Extension
- **Docker Machine Azure Driver**
- **Docker for Windows vs. Docker on Windows**
 - Former from Docker Inc., latter from Microsoft Corp.
- **Docker Registry Azure Storage Driver**
- **Azure Container Service (ACS)**
 - For Linux & Windows (preview)
 - Orchestrators: Swarm, Mesos
- **ARM Templates**
 - Single VM deployments, ACS clusters
- **Azure File Service Docker Volume Driver**
 - Currently only for Linux, Windows coming



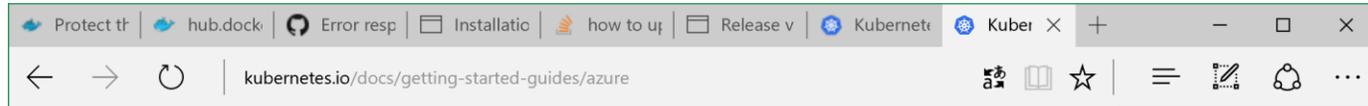
Kubernetes

- Can be used to manage a cluster of containers
 - A single master can manage one or more Docker hosts called nodes
 - Dynamically allocates containers on to hosts
- Groups containers into logical units
 - “Labels” and “pods”
- Pod and service definitions are simply JSON files



<https://github.com/kubernetes/kubernetes/blob/master/docs/getting-started-guides/coreos/azure/README.md>

Kubernetes on Azure



A screenshot of the Kubernetes.io website. The page title is 'Guides' and the subtitle is 'How to get started, and achieve tasks, using Kubernetes'. There are navigation tabs for 'GUIDES', 'REFERENCE', 'SAMPLES', and 'SUPPORT'. A search bar is present. On the left, there is a 'Guides' sidebar with a tree view of topics: Getting Started, Accessing the Cluster, Workload Deployment and Management, Batch Jobs, Service Discovery and Load Balancing, Containers and Pods, Monitoring, Logging, and Debugging Containers, and Creating a Cluster. The main content area shows the title 'Running Kubernetes on Azure (Flannel-based)' and a list of prerequisites: Prerequisites, Cluster operations (Cluster bring-up, Cluster deployment examples), Cluster scaling, Cluster tear-down, and Notes. A 'Prerequisites' section is highlighted at the bottom.

A screenshot of a Microsoft Developer blog post. The title is 'Kubernetes Cluster automated deployment on Azure – First Step'. The author is Stéphane GOUDEAU, dated July 11, 2016. The post includes social media share buttons for Facebook (0), Twitter (0), and LinkedIn (10). The main text starts with 'Last week was published a video which I think may be usefully completed by a few articles that gives more details ...'. It mentions a video by Hervé Leclerc (Alter Way CTO) and an Open Source implementation. A list of planned articles follows: 'Kubernetes and Microsoft Azure', 'Programmatic scripting of Kubernetes deployment', 'Provisioning the Azure Kubernetes infrastructure with a declarative template', 'Configuring the Azure Kubernetes infrastructure with Bash scripts and Ansible tasks', 'Automating the Kubernetes UI dashboard configuration', and 'Using Kubernetes...'. The post concludes with 'Let's start first by presenting Kubernetes and why and how to deploy it in Azure.' and a link to 'Kubernetes'.

Q&A



www.futuredecoded.it

 #FutureDecoded