



2018

Global Azure BOOTCAMP

Verona



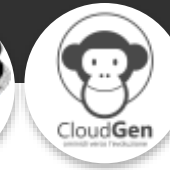


24 / CO

Platinum Sponsor



Gold Sponsor



Basic Sponsor

Tweet della giornata



#GlobalAzure

@clouddgen_verona



2018
Global Azure
BOOTCAMP

ARGOMENTO

I Containers in Azure light-VM o un vero cambio di paradigma?

Iniziando dalla definizione di container, nella sessione esamineremo quali strumenti Microsoft Azure mette a disposizione per sfruttare pienamente la potenza e flessibilità offerte da Docker nella containerizzazione dei processi. Partendo da una infrastruttura semplificata a nodo singolo su vm, arriveremo ad allestire un servizio ACS (AKS) per il clustering con orchestration in Microsoft Azure, predisponendo una infrastruttura su cui sperimentare scaling&orchestration di progetti Docker.

I Containers in Azure, light-VM o un vero cambio di paradigma?



GIULIANO LATINI



@giulianolatini



giulianolatini



giulianolatini



About me

ITPro > 25 year

Senior Software Developer at INNOTEAM s.r.l.

Linux Foundation Member

CTS Senior Member at Federation IDEM AAI

ICT Administrator at Università Politecnica delle Marche

Istruttore Informatico at Comune di Chiaravalle (AN)

Member of WindowServer.it community

Member of DevMarche community



Agenda

- Container or not Container, this is the trouble
- Docker, Containers make easy
- Microsoft Azure, What's The Containers?
- Write Infrastructure as if there were no tomorrow
- VM, Single Node Approach
- AKS/ACS, cluster and orchestration together
- AKS, Deploy it, Use it



“

Standardizing the Shipping of Software

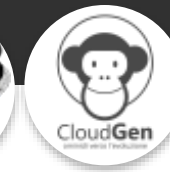
The concept of container technology uses the paradigm of shipping containers in inter-modal transport.



“

Proponents of container technology have pointed out that by sharing an OS and eliminating the need for VMs, containers consume far fewer resources than systems using VMs or hypervisors.

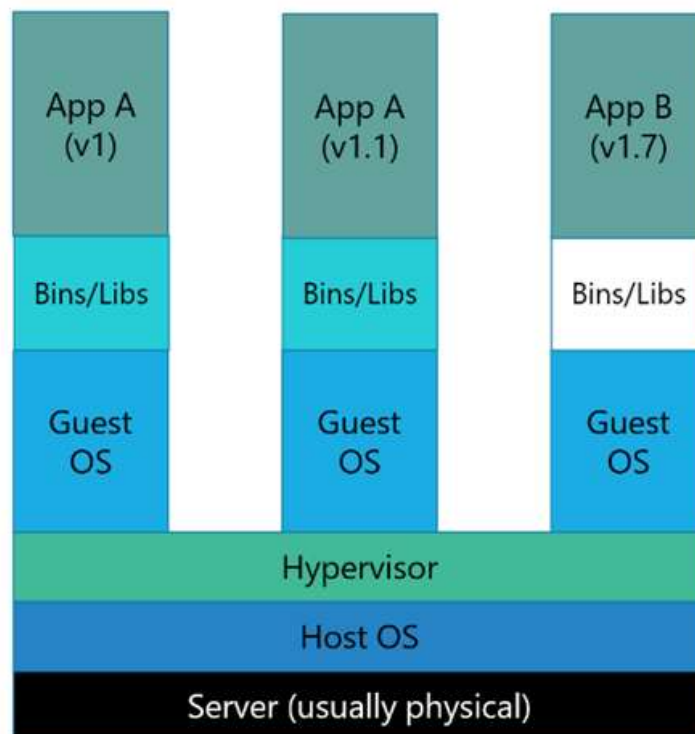
I Containers in Azure, light-VM o un vero cambio di paradigma?



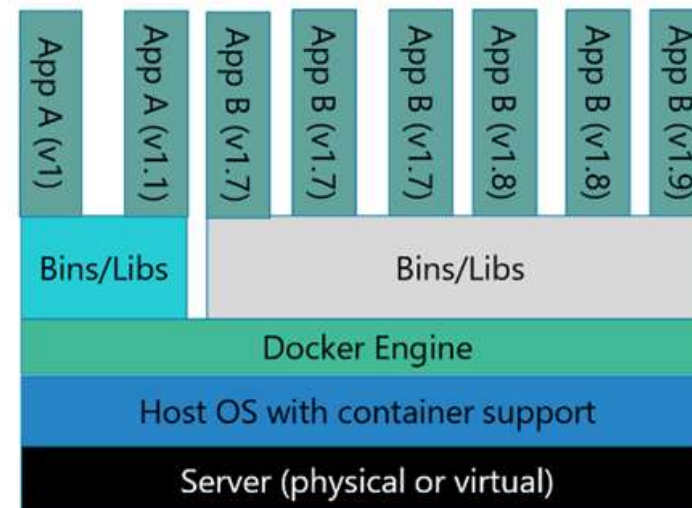
Container or not Container, this is the trouble



Server Virtualisation: Each app and each version of an app has dedicated OS



Containers: All containers share host OS kernel and appropriate bins/libraries



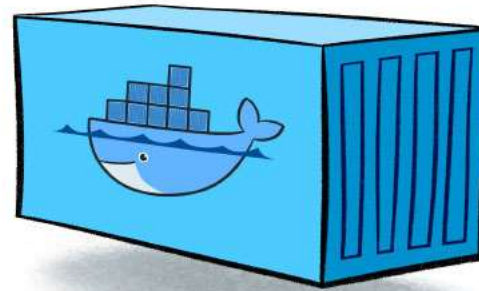
VM vs. Docker

Size		
Startup		
Integration		

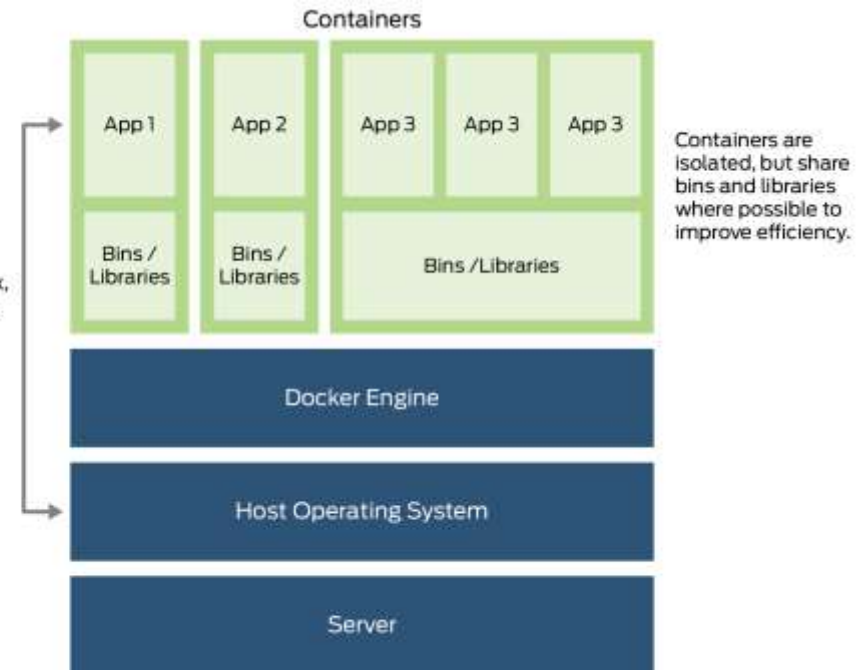
I Containers in Azure, light-VM o un vero cambio di paradigma?



Docker, Containers make easy

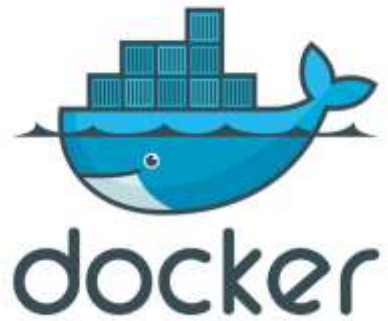


Containers are created with Linux, but share a kernel with almost any type of host OS.

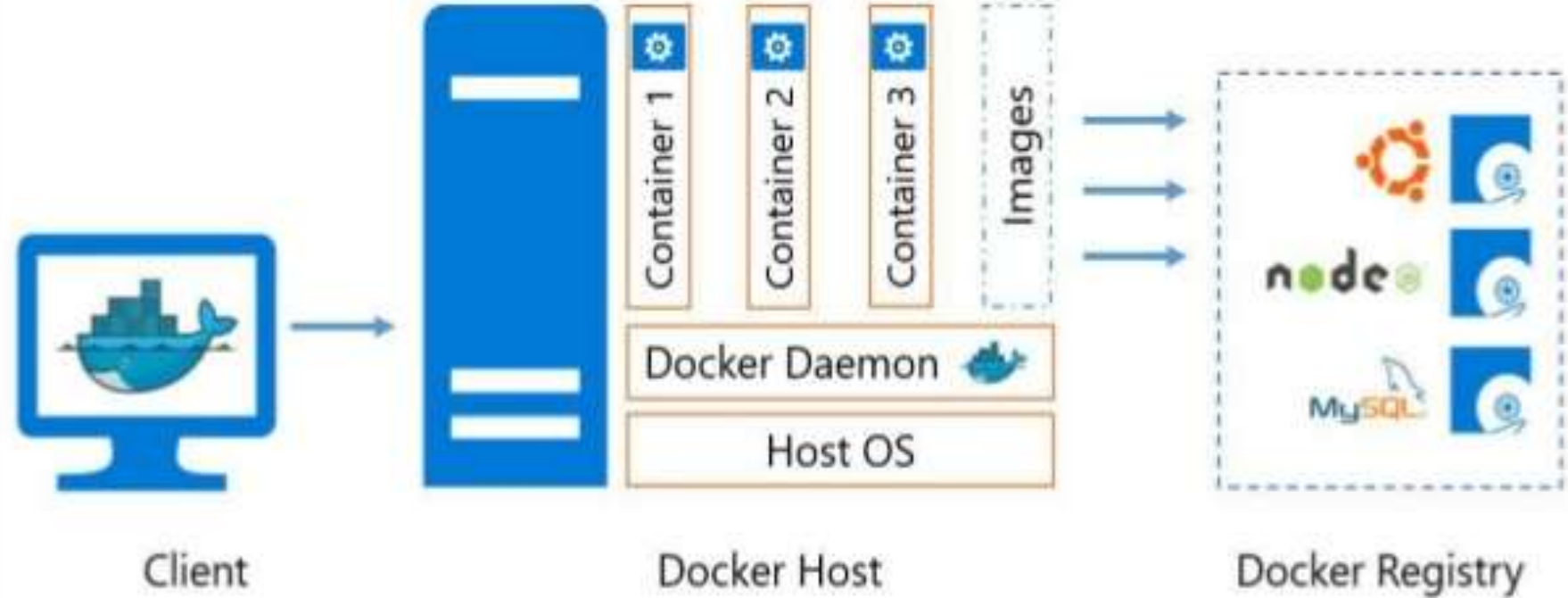


I Containers in Azure, light-VM o un vero cambio di paradigma?

Docker, Containers make easy



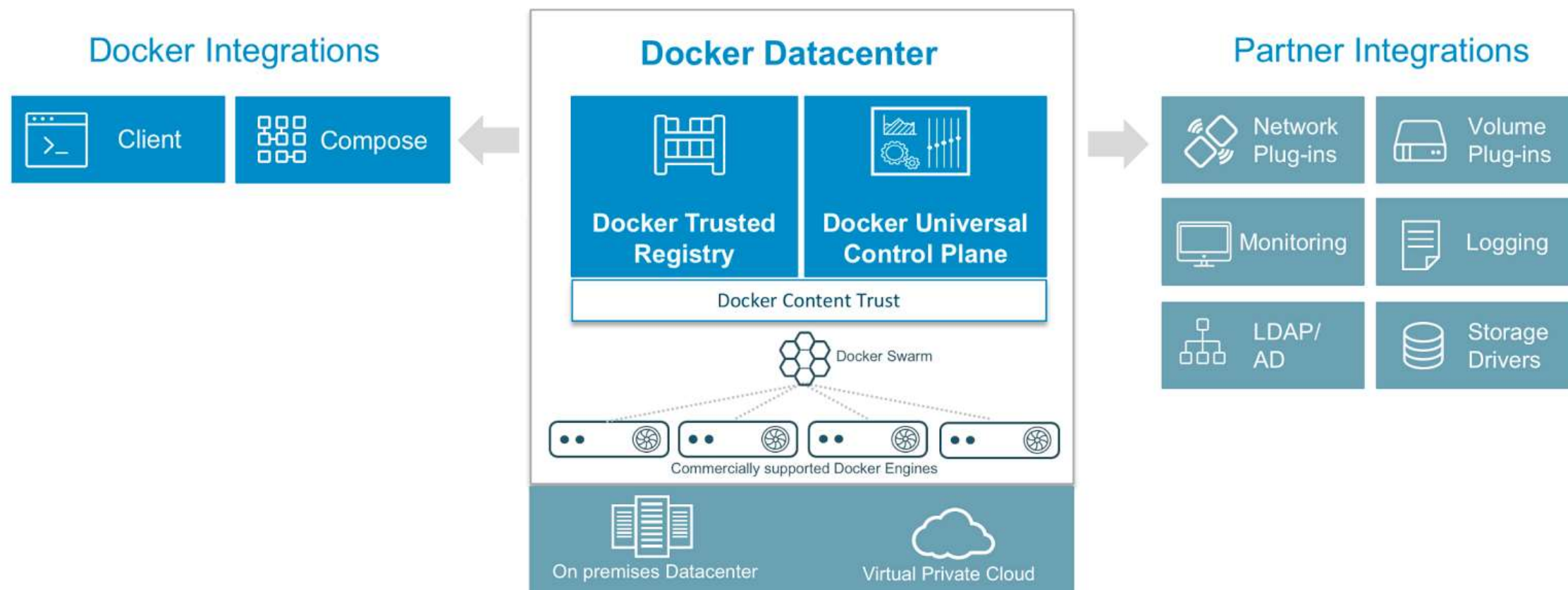
Docker's Architecture



I Containers in Azure, light-VM o un vero cambio di paradigma?



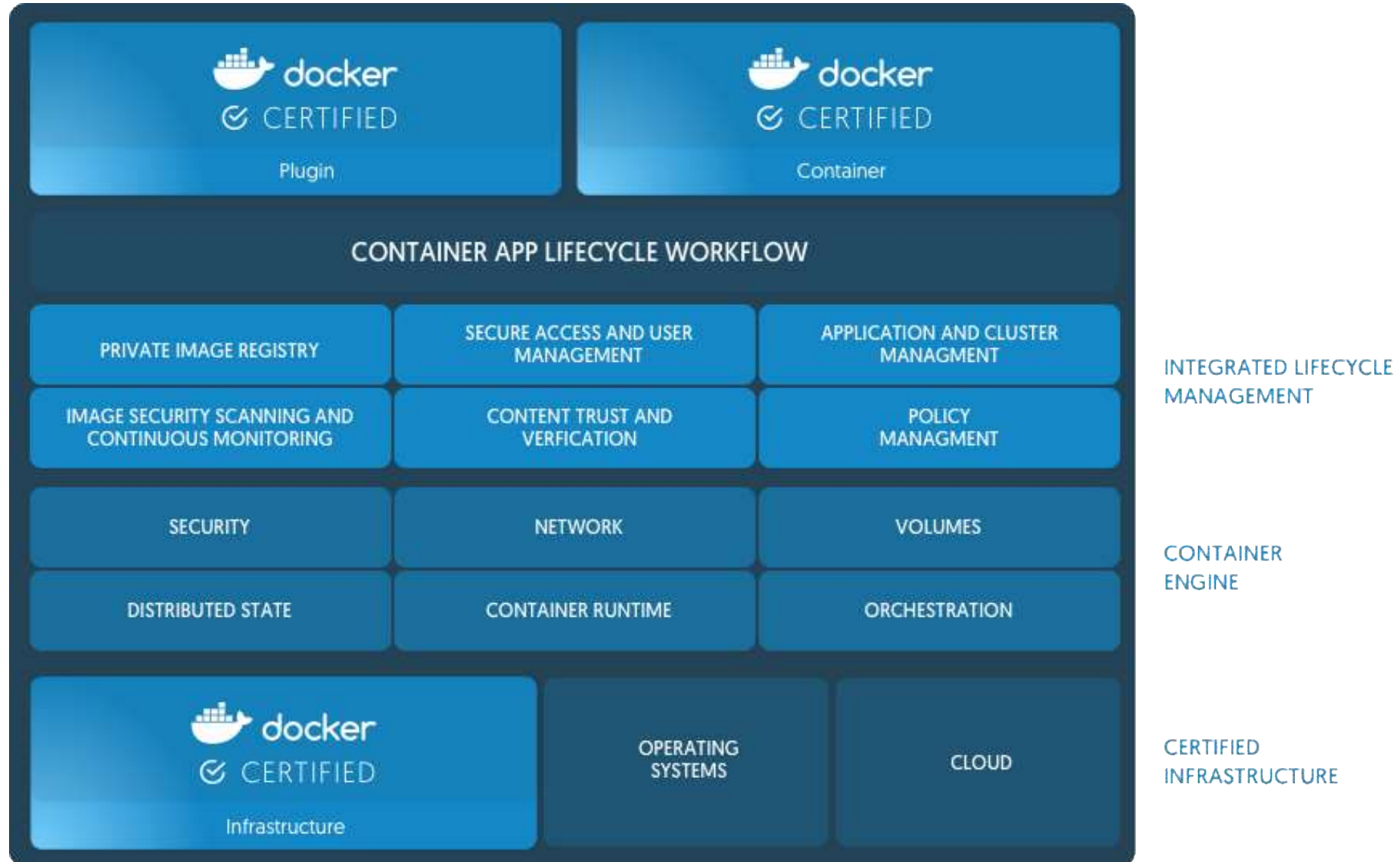
Docker, Containers make easy



I Containers in Azure, light-VM o un vero cambio di paradigma?

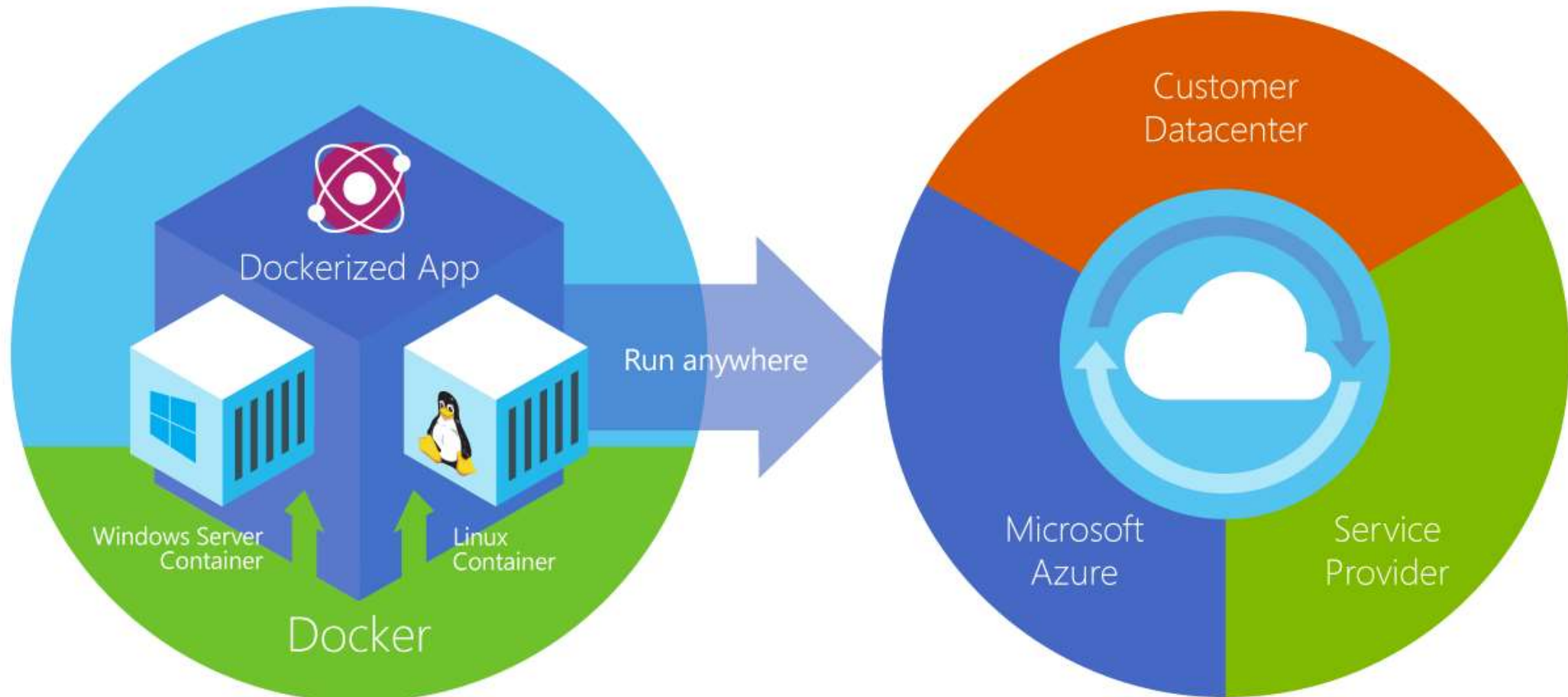


Docker, Containers make easy



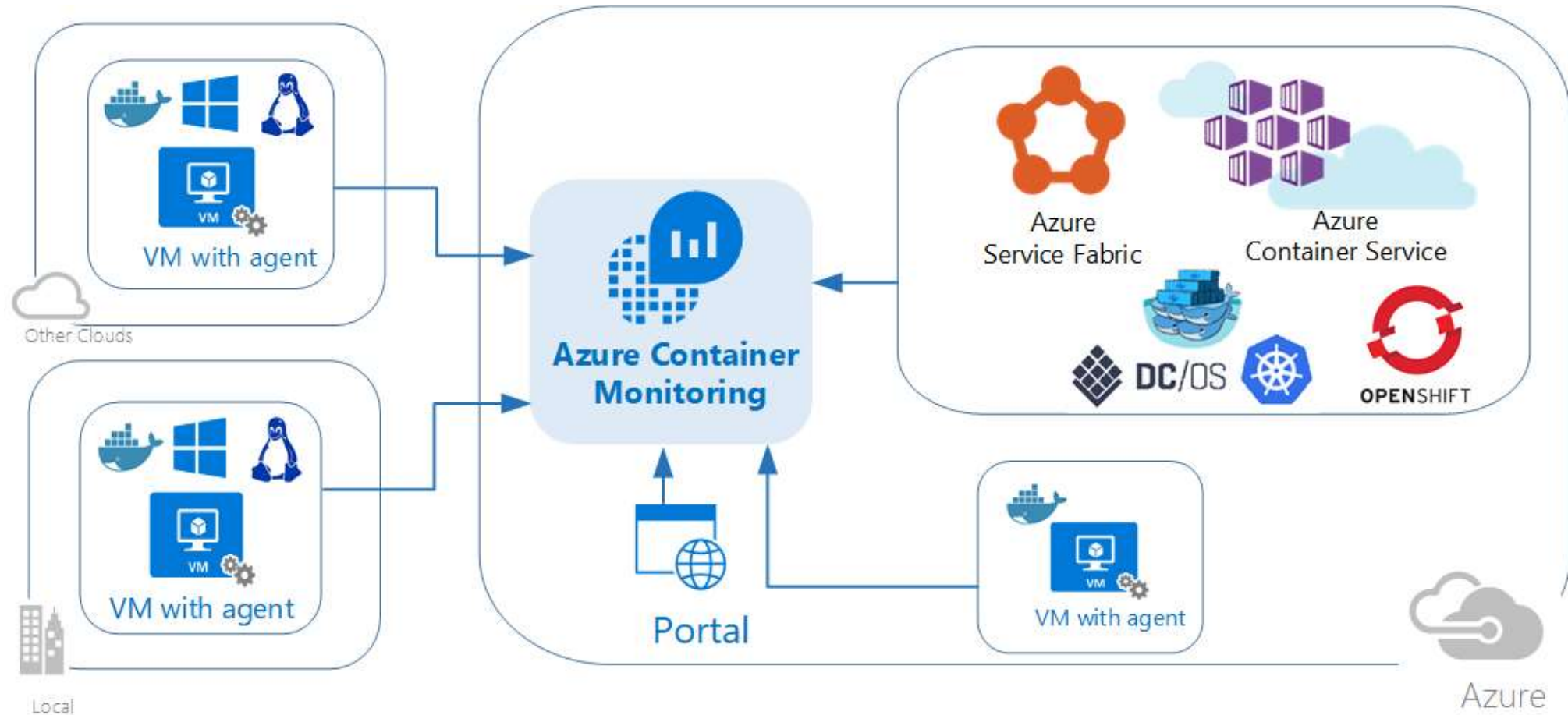
I Containers in Azure, light-VM o un vero cambio di paradigma?

Microsoft Azure, What's The Containers?



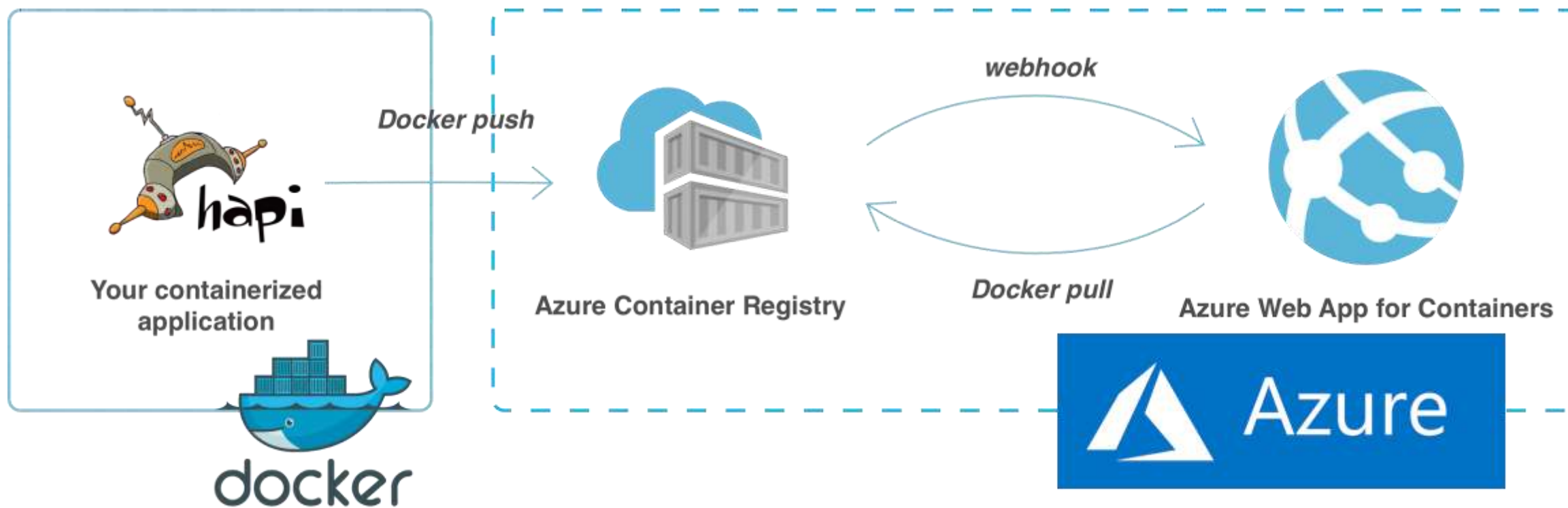
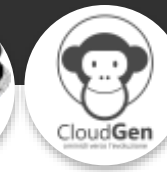
I Containers in Azure, light-VM o un vero cambio di paradigma?

Microsoft Azure, What's The Containers?



I Containers in Azure, light-VM o un vero cambio di paradigma?

Microsoft Azure, What's The Containers?



I Containers in Azure, light-VM o un vero cambio di paradigma?



Write Infrastructure as if there were no tomorrow

```
docker-compose.yml x
1 version: '3'
2
3 services:
4   bookings-api:
5     image: smarthotels/bookings
6     build:
7       context: .
8       dockerfile: SmartHotel.Services.Bookings/Dockerfile
9
10  hotels-api:
11    image: smarthotels/hotels
12    build:
13      context: .
14      dockerfile: SmartHotel.Services.Hotels/Dockerfile
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

~/source/SmartHotel360-Azure-backend/src \$:

EXPLORER

- OPEN EDITORS
 - Dockerfile
- NODE-TODO
 - .vscode
 - app
 - config
 - node_modules
 - public
 - .gitignore
 - ! docker-compose.debug.yml
 - ! docker-compose.yml
- Dockerfile
- license
- package.json
- README.md
- server.js
- yarn.lock

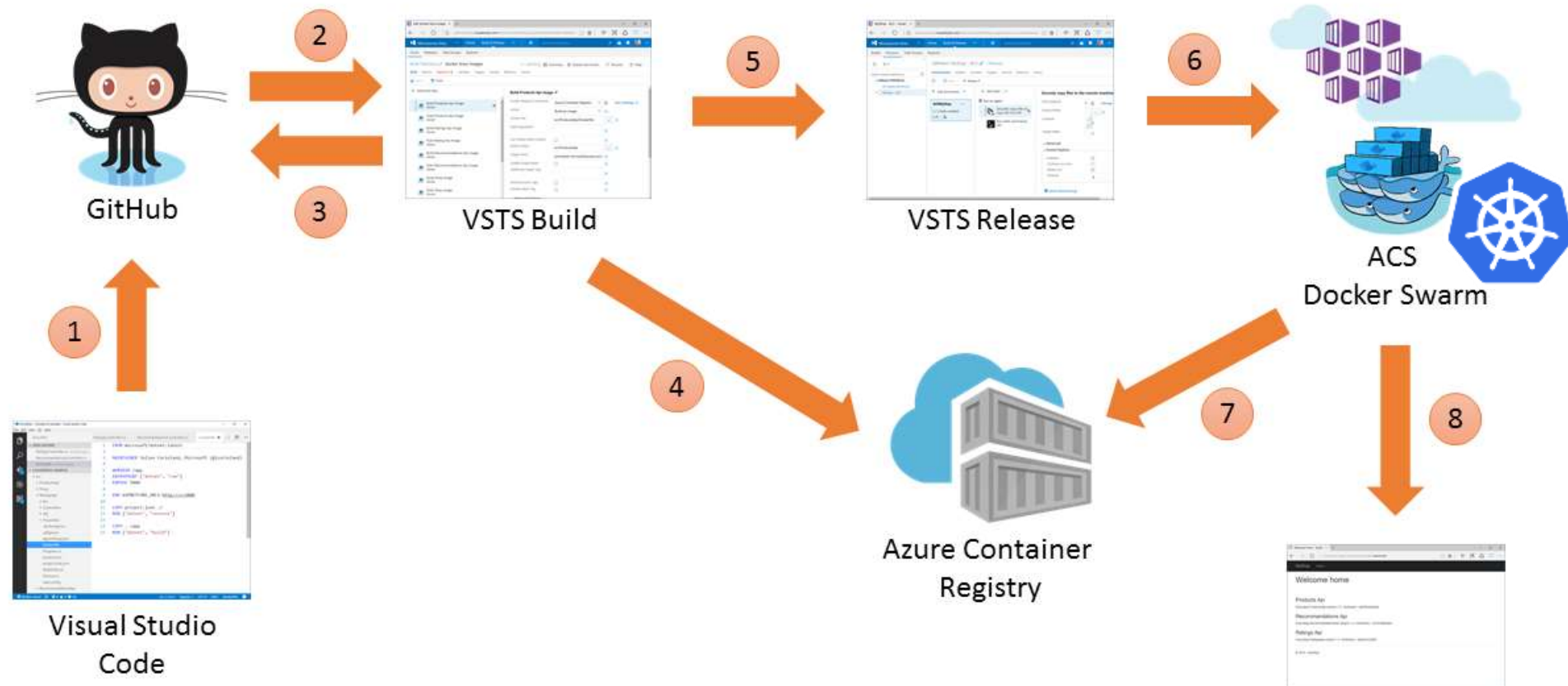
Dockerfile x

```
1
2 FROM node:latest
3 MAINTAINER Scotch
4 LABEL Name=node-todo Version=0.0.1
5 COPY package.json /tmp/package.json
6 RUN cd /tmp && npm install --production
7 RUN mkdir -p /usr/src/app && mv /tmp/node_modules /usr/src
8 WORKDIR /usr/src/app
9 COPY . /usr/src/app
10 EXPOSE 8080
11 CMD node server.js
12
```

```
PS C:\dev\netcore-webapi-microservice-docker> docker build -t cesardl/netcore-webapi-microservice-docker:first .
Sending build context to Docker daemon 1.148 MB
Step 1 : FROM microsoft/dotnet:latest
latest: Pulling from microsoft/dotnet
5c90d4a2d1a8: Downloading [=====>] 18.34 MB/51.35 MB
ab30c63719b1: Downloading [=====>] 18.48 MB/18.55 MB
c6072700a242: Downloading [=====>] 18.34 MB/42.53 MB
121d7eef6c20: Waiting
eb57cf4f29ee: Waiting
b2c5ae2d325b: Waiting
```

I Containers in Azure, light-VM o un vero cambio di paradigma?

Write Infrastructure as if there were no tomorrow

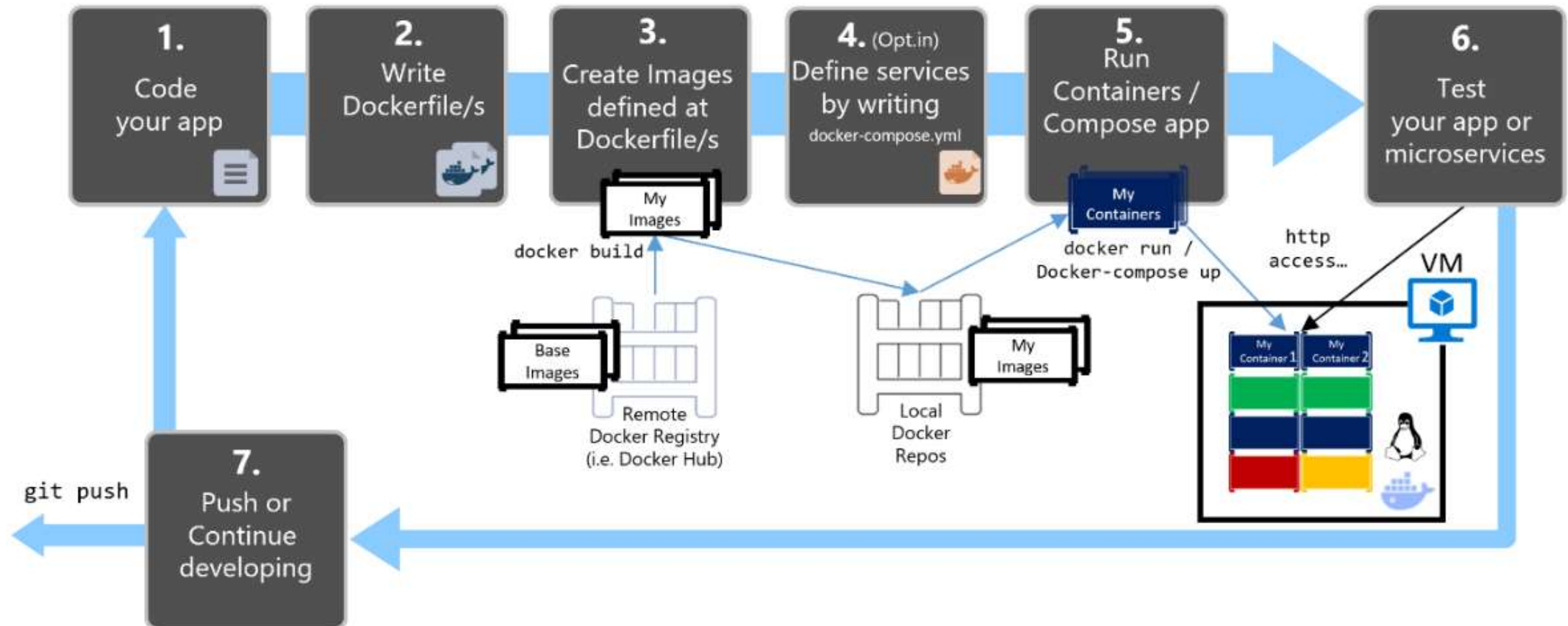


I Containers in Azure, light-VM o un vero cambio di paradigma?

VM, Single Node Approach

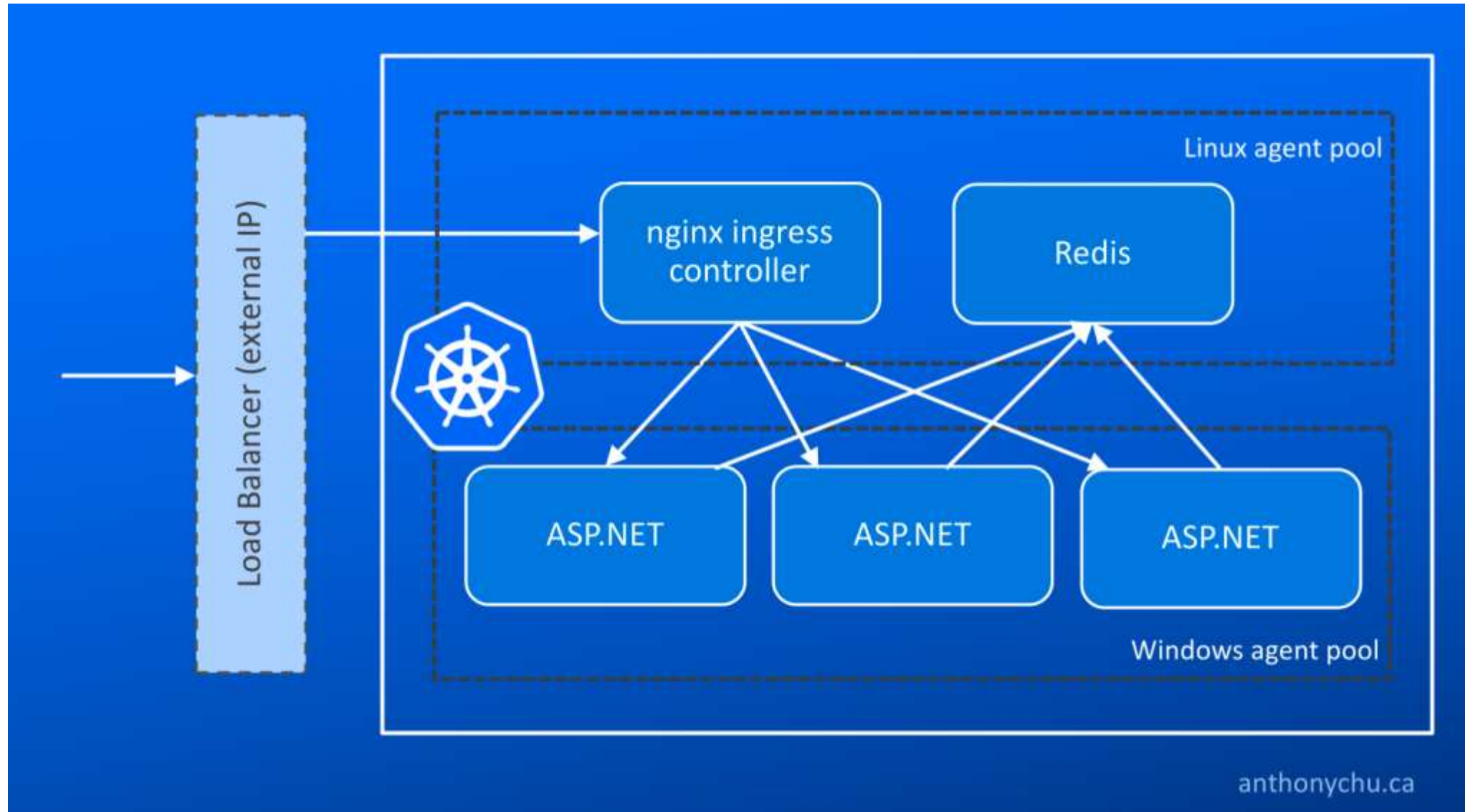


Inner-Loop development workflow for Docker apps



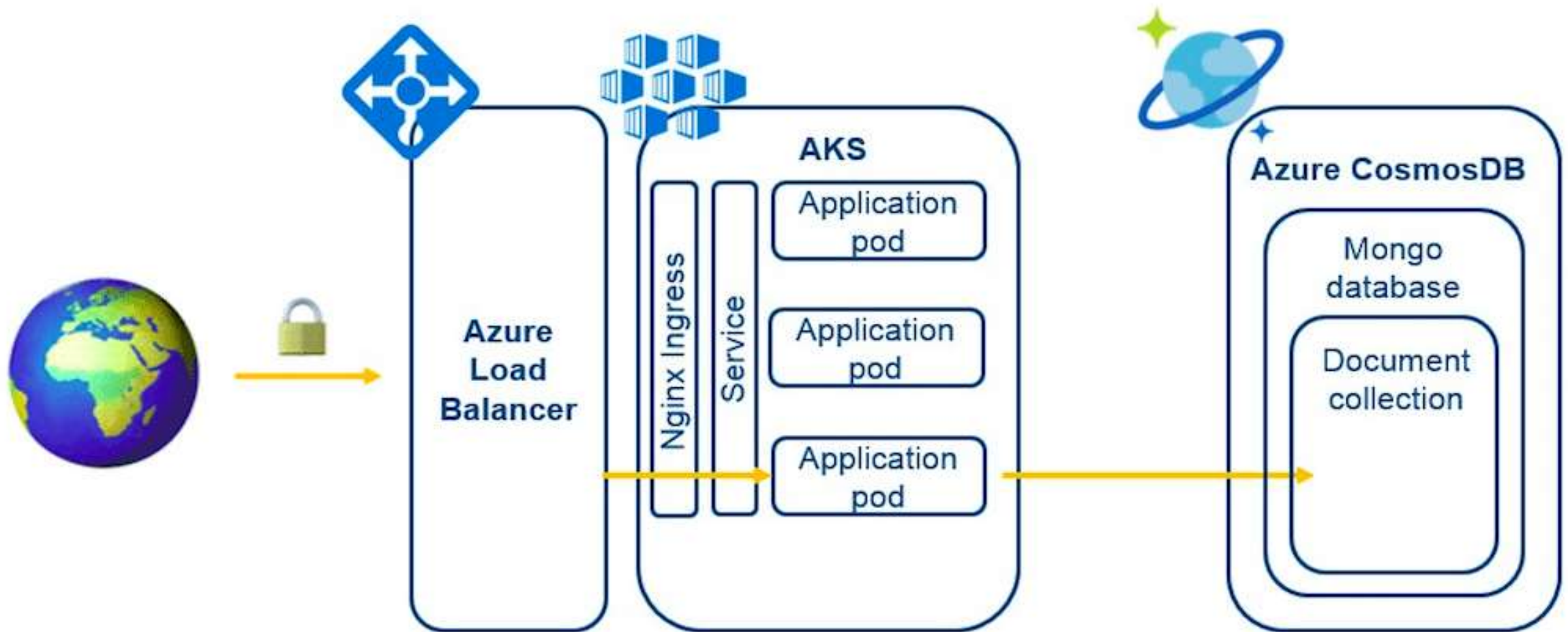
I Containers in Azure, light-VM o un vero cambio di paradigma?

AKS/ACS, cluster and orchestration together



I Containers in Azure, light-VM o un vero cambio di paradigma?

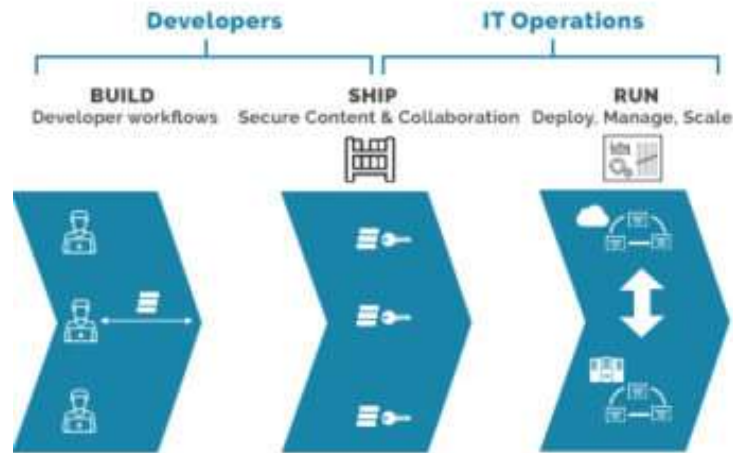
AKS/ACS, cluster and orchestration together



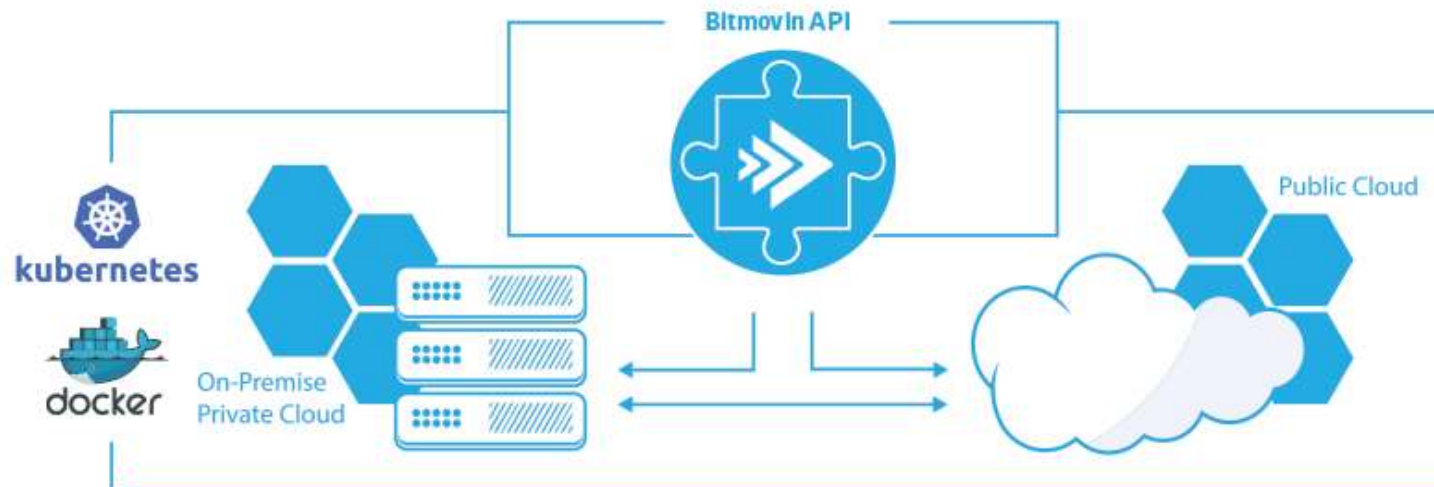
I Containers in Azure, light-VM o un vero cambio di paradigma?



AKS, Deploy it, Use it



```
apiVersion: v1
kind: Service
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # if your cluster supports it, uncomment the following to automatically create
  # an external load-balanced IP for the frontend service.
  loadBalancerIP: 52.179.14.59
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    app: guestbook
    tier: frontend
```





Demo

Grazie

Domande?



giulianolatini



@giulianolatini



giulianolatini