



Il cloud oltre la virtualizzazione

GARR Workshop, Aprile 2017

Agenda



- Infrastructure & Operations
- The world is hybrid
- Density and elasticity
- Kubernetes case
- Conclusions
- Questions



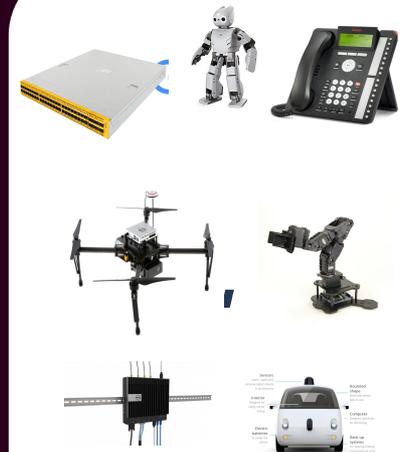
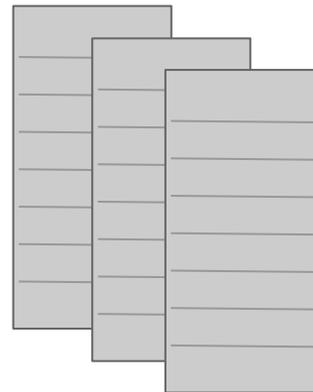
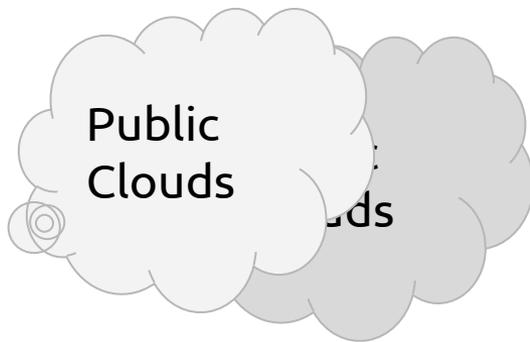
Cloud

Elastic hybrid cloud

Edge

Transactional Edge (IoT)

Bulk compute data center





Cloud

Elastic hybrid cloud

Edge

Transactional Edge (IoT)

Public cloud | Private cloud | Containers | Serverless

Edge of network /IoT



Logos for cloud providers and services including: NETFLIX, Instagram, Dropbox, Windows Azure, amazon web services, Google Cloud Platform, IBM, rackspace, ORACLE PUBLIC CLOUD, and Joyent.

Logos for telecom and enterprise companies including: Walmart, AT&T, Deutsche Telekom, Time Warner Cable, Cisco webex, NTT, CISCO, WELLS FARGO, YAHOO! JAPAN, rackspace, NEC, peer1, Bloomberg, ebay, HM Government, SAMSUNG, SAMSUNG SDS, hp, COMCAST, sky, and LexisNexis.

Logos for IoT and hardware companies including: TESLA, Google, DELL, SAMSUNG, GE, IBM, intel, DJI, QUALCOMM, Parrot, BROADCOM, and Open Source Robotics Foundation.

Infrastructures & Operations



Public Cloud

- Amazon
- Azure
- Google
- Rackspace

Less Operations overhead,
higher costs and you have to
trust the operator

Private Cloud

- Data Center/Bare Metal
- OpenStack
- VMWare
- Kubernetes
- Mesos
- Swarm

More Operations overhead,
data in house and other
benefits

TCO is driven by operations



Operations

Cost

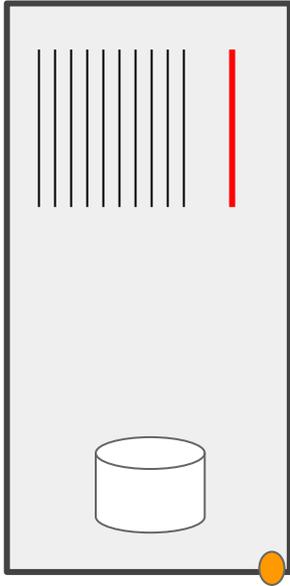
Licenses

Density & Elasticity



- Resources elasticity
- Virtual nodes/Hardware nodes ratio, increase
- Nodes/Humans ratio, increase

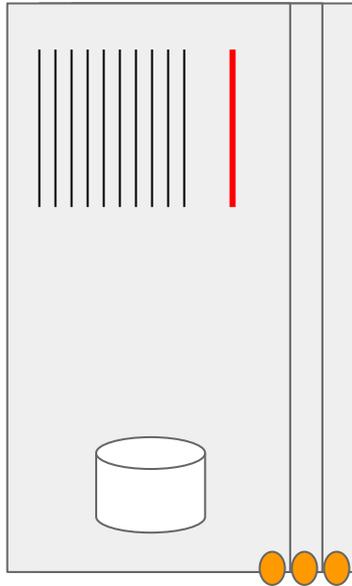
Virtual Machines



Traditional operations

KVM

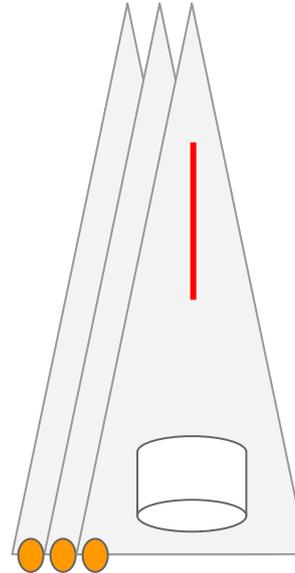
Machine Containers



Traditional operations

LXD

Process Containers



Mesos / **Kubernetes** / Swarm

Docker / RunC / Rkt / OCID

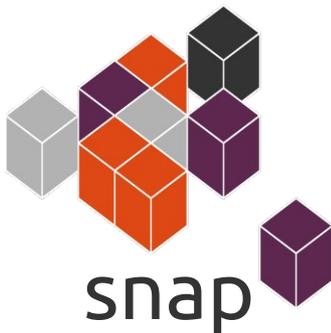
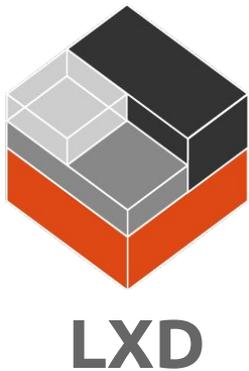
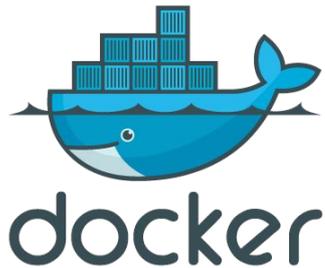
Snap Application Containers

Snap extends host



Host Linux Filesystem

Process, Machine, & Application Containers



- **Docker** *runs process* containers
 - Contains at most one process or “service” active per container
 - Shared kernel, own filesystem/network
- **LXD** *boots machine* containers
 - Launches /sbin/init + sshd + syslog + cron + atd + ...
 - Shared kernel, own filesystem/network
- **Snap** *provides application* containers
 - Packages apps to run on the native system, but secured
 - Shared kernel/filesystem/network

Kubernetes Story



Lesson learnt

1

AppEngine
launched in 2011
(2008) - serverless

2

Few years later
GCE

3

In 2015
Kubernetes has
been released
Recently GKE

Kubernetes Story



What's the strategy ?

- Hire people already trained to understand a complex and distributed infrastructure
- Train the users and the market writing applications in a distributed way

Kubernetes Application Model



Hardware



Single Host



Bare Metal Cluster



Cloud Cluster



Cloud

Microsoft Azure



Single Host



Cloud Cluster





- Always **pure**, upstream, latest, greatest kubernetes
- 100% **compatibility** with Google's Kubernetes
- **Easily deploys** on AWS, Azure, GCE, OpenStack, and metal
- Rich, growing **ecosystem** of smart accessories
- **Cost effective** at scale



Microsoft Azure



Conclusions



What's the challenge in the future ?

- Think in a distributed way
- Write applications for a distributed environment, not for virtualization
- Do not rely on the infrastructures, be hybrid
- Model and automate everything



Questions ?

GARR Workshop, Aprile 2017

Paolo de Rosa
paolo.de.rosa@canonical.com





ubuntu

Supported by Canonical