#6162 - How to Build and Operate Your Private Cloud with Ubuntu on IBM Power Systems and OpenPOWER

Ivan Doboš, Solutions Architect, IHV team
Canonical

Think Conference
March 2018
Delivering the economics and agility of hyperscale innovators to enterprises globally

Developer productivity | Infrastructure economics
IBM / Canonical Power Relationship

● 14.04 LTS first release (April 2014)

● First commercial Linux to support:
  ■ POWER8 LE
  ■ CAPI
  ■ KVM, PowerKVM
  ■ Docker, LXC, LXD
  ■ Cloud Foundry

● Canonical is Platinum OpenPower Member
  ○ Robbie Williamson - Chairman of the Board)
  ○ Ivan Doboš - Technical Steering Committee

● Ubuntu Open Source ecosystem available for Power; IBM software charms available now
Why cloud?

- (Hyper)scale
- Flexibility
- Efficiency ==> better economics

- Successful cloud
  - Hosting applications
  - Growing (scaling out)
  - Providing new features (Regular upgrades)
  - minimizing operational costs by Automation
From bare metal to containers ...

- Hardware (compute, network, storage)
- Operating system
  - Applications
  - Containers
    - Applications
    - Containers *
      - Applications
  - Hypervisor
    - Virtual Machines
      - Operating System
      - Applications
      - Containers
        - Application

* nested containers, Docker under LXD example
Cloud Solution Stack: Enabling Operations at Scale

**Landscape:**
- Systems management & patching

**LXD:**
- Pure-container hypervisor

**Ubuntu OpenStack:**
- Canonical-produced optimized and interop tested openstack packages
- Enterprise-class, hyperscale server operating system

**Ubuntu Server**

**VMs - KVM**

**Containers (LXD)**

**Juju**
- Cloud deployment, integration, scaling, upgrading

**MAAS:**
- Metal-as-a-Service for bare-metal provisioning

**OpenStack**

**Containers (Docker)**

**Kubernetes:**
- Container coordination

**Docker**
First things first – Bare metal

maas.io

Key to Private DC Economics is Automation
Metal as a Service
Say hello to your physical cloud

Total automation of your physical servers for amazing data centre operational efficiency. On premise, open source and supported.

- Zero-touch deployment of Windows, Ubuntu, CentOS, RHEL and SUSE
- Catalogs and manages every device in every registered server
- Inventory management of every device in every server
- Remote setup of NIC bonding and RAID storage
- Open-source IP Address Manager (IPAM)
- Automate everything from IPMI and PXE to fully converged chassis, including Cisco UCS, HP Moonshot and more
- Integrate with devops automation tools like Juju, Chef, Puppet, SALT, Ansible and more
- Automatically enable remote access through SSH and Powershell
- REST API, Web UI and CLI
MAAS Feature Matrix

SDDC
- IPAM, DNS and DHCP
- Dynamic allocation
- Availability Zones
- SSH & BMC credential setup
- Auto-enlistment via PXE
- Hardware inventory
- Auto-tagging
- Custom commissioning
- Post-install configuration

User Interfaces
- Web
- REST API
- CLI
- Python

Operating Systems
- CentOS 6, 7
- RHEL 6, 7
- Ubuntu
- Windows 2003, 2010
- Custom

Disk layout
- Partitioned
- Raw
- RAID
- LVM
- bcache

BMCC protocols
- IPMI
- AMT
- Redfish

Chassis support
- Cisco UCS
- Microsoft OCS
- Seimicro 15K
- HP Moonshot

VM Controllers
- KVM Virsh
- PowerKVM
- vSphere

Host NIC config
- Raw
- Tagged VLAN
- Bonded
- Bridged

SDDC
- IPAM, DNS and DHCP
- Dynamic allocation
- Availability Zones

User Interfaces
- Web
- REST API
- CLI
- Python

Operating Systems
- CentOS 6, 7
- RHEL 6, 7
- Ubuntu
- Windows 2003, 2010
- Custom

Disk layout
- Partitioned
- Raw
- RAID
- LVM
- bcache

BMCC protocols
- IPMI
- AMT
- Redfish

Chassis support
- Cisco UCS
- Microsoft OCS
- Seimicro 15K
- HP Moonshot

VM Controllers
- KVM Virsh
- PowerKVM
- vSphere
Time from request to VM
>>> from boto import ec2
>>> c = ec2.connect('us-west-2')
>>> r = c.run_instances(
    min_count=3,
    aws_image_id,
    instance_type='c1.xlarge')
# The devil in the details

## Node
- Discovery
- HW inventory
- BMC
- BIOS & firmware
- Boot process
- OS Deployment
- Diskless / Ephemeral
- Storage & Networking
- Decommissioning

## PXE & TFTP
- lshw
- freeipmi
- fwupd
- BIOS & UEFI
- curtin
- iSCSI
- cloud-init
- NIST-wipe

## Network
- DNS, NTP & more
- Topology & addressing
- L2 & L3 config

## Infrastructure
- PDUs
- Switches & routers
- Storage appliances

## Network
- bind, ntpd
- IPAM
- DHCP
- DCIM
- OpenFlow
- SNMP MIBs
- Vendor stuff
MAAS
The bare metal API
Inventory. Commission. Deploy.
IP, DHCP, DNS, PXE, IPMI ready to roll
1. Automated **physical provisioning**
2. **Dynamic allocation** to workloads
3. **IP Address Management** (IPAM)
4. Web interface and **REST API**
5. **Windows, Linux** OS install
"Please give me up a machine of ppc64el architecture, with 24 GB RAM, a root disk of 300GB and at least 6 more 3TB disks, with an nVidia GPGPU running Ubuntu."
MAAS 2.0
Changes the provisioning game
100k servers in a Region

**HA Region Controller**
(1-3 servers)
- highly available
- users and groups
- resource allocation
- central postgres db
- machine inventory

Single view of server status, API endpoint, web UI.

**Rack Controller**
- * ideally on switch
- * typically in-rack
- * cache of images
- * backup for adjacent racks

Provides PXE boot and local OS image delivery for installation.

**Server**

**Server**

**Server**

**Server**

**Server**

**Server**
High Availability for Region & Rack services
ubuntu.com/lxd
github.com/lxc
linuxcontainers.org
LXC 2.0 adds the LXD container hypervisor
LXD provides machine containers
Physical Machines

Virtual Machines

Linux Containers

Machine containers

Physical Machines
LXD is Canonical’s container hypervisor

- Ultra fast “vm-lite” guests (bare metal speed)
- Any distribution of Linux - e.g. Ubuntu, CentOS
- Starts in less than 1 second
- 15x density of KVM or ESX for idle workloads

LXD REST API

nova-lxd
lxc cli
other restful apps
LXD – benefits for Cloud

- Lift and shift for existing applications
- Collocation of different workloads on the same node
  - Hyper-converged architecture as a default for Canonical OpenStack
- More efficient way of using instances in cloud
  - Typically an instance in Cloud = VM
  - Run containers inside an instance
- Instances in cloud implemented as LXC containers
  - LXD as a supported hypervisor for OpenStack
  - 14x greater density with LXD than with VMs (tested on x86), lower latency
Operate big software at scale on any cloud

juju

jujuucharms.com
the phase change of modern software
scarcity has shifted from code to ops

(Are you ready to operate a SW you don’t understand?)
TCO is driven by operations
How to fix it?

- Imagine ‘open source operations’
  - Free, downloadable, driven by the community and/or SW vendors
  - Reflecting the release changes of the applications to operate
  - Providing functions like:
    - installation/uninstallation
    - upgrades
    - configuration management
    - scale
    - integration between applications
    - ‘must to have’ generic and also application specific actions: backup, restore, pause/restart, ...
  - reuse operations code in completely different architectures
Reuse across clouds

- developer laptop (lxd containers)
- test and dev vm’s
- vmware
- bare metal
- openstack

- gce / azure / oracle
- aws
Ubuntu OpenStack

65% of large production OpenStack deployments run Ubuntu
Canonical OpenStack

Management & Automation

- Landscape
- Juju

Infrastructure Services

- Nova
- Horizon
- Ceilometer / Telemetry
- Keystone
- Neutron
- Swift
- Cinder
- Glance

Ubuntu 16.04 LTS

MAAS

Intel

Power

IBM LinuxONE
OpenStack Archive Upgrade Guarantee

- Ubuntu 14.04 LTS: 5 years
- Ubuntu 16.04 LTS: 5 years
- Ubuntu 18.04 LTS: 5 years
- Mitaka LTS: 5 years
- Queens LTS: 5 years
- Juno: 18 months
- Liberty: 18 months
- Kilo: 18 months
- Extra 18 months for customers
- Pike: 18 months
- Extra 18 months for customers
- Ocata: 18 months
- Newton: 18 months
Canonical’s distribution of Kubernetes
easy, stable, latest release of upstream
any CPU architecture

x86

s390x
extensible by design

- Calico
- Weave
- Flannel
- Ceph
- NFS
- NVIDIA
rich accessory ecosystem

weavecloud
ZABBIX
sysdig
elastic
GALACTIC FOG
PROMETHEUS
Nagios®
RANCHER
wercker
Kubeless
fabric8
Why Canonical?

- Most popular cloud and container operating system
- Launch partner as a KCSP
## Why Canonical Kubernetes?

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pure upstream</strong></td>
<td>Latest &amp; greatest versions</td>
</tr>
<tr>
<td><strong>Operates</strong></td>
<td>on AWS, Azure, GCE, OpenStack, VMWare</td>
</tr>
<tr>
<td><strong>Bare metal</strong></td>
<td>operations with MAAS</td>
</tr>
<tr>
<td><strong>100% compatible</strong></td>
<td>with Google’s Kubernetes</td>
</tr>
<tr>
<td><strong>Secured</strong></td>
<td>TLS, Kernel Live patching, confinement</td>
</tr>
<tr>
<td><strong>Upgradable</strong></td>
<td>between each Kubernetes Release</td>
</tr>
<tr>
<td><strong>Cost effective</strong></td>
<td>at scale</td>
</tr>
</tbody>
</table>
Support for GPUs

- Automated GPU Discovery
- Live GPU adjustments
- Automated configuration of applications
LXD, Kubernetes & GPUs

NVidia CUDA inside a LXD container
By Stéphane Graber on 28 March 2017

GPUs and Kubernetes for deep learning—
By Samuel Cozannet on 15 February 2017

How we commoditized GPUs for Kubernetes
By Samuel Cozannet on 19 April 2017

[Edit 2017-04-20] A careful reader informed me (thanks for that HN user puzzle) that it is no longer required to run in privileged mode to access the GPUs in K8s. I therefore removed a note that previously stated this requirement, and am in the process of updating my Helm charts to remove it as well from there.

Over the last 4 months I have blogged 4 times about the enablement of GPUs in Kubernetes. Each time I did so, I spent several days building and destroying clusters until it was just right, making the experience as fluid as possible for adventurous readers.

It was not the easiest task as the environments were different (cloud, bare metal), the hardware was different (g2.xlarge have old K20s, p2 instances have K80s, I had 1080CTX at home but on consumer grade Intel NUC...). As a result, I also spent...
Conclusion

Juju + helm, sudo give me Artificial Intelligence: check!

*coming soon to IBM Power
In Summary

➢ Ubuntu #1 Linux for Cloud, ScaleOut, Dev Ops
  - >65% Production OpenStack clouds are Ubuntu
  - Juju and related cloud tooling supports DevOps speed

➢ Operating on premise with the efficiency and flexibility of a public cloud
  - Combine maas, LXD, Juju and OpenStack as needed

➢ Ubuntu on IBM Systems: great Ubuntu experience on IBM Power and IBM Z/LinuxONE
  - Global Partnership with IBM
We are Canonical

It is our mission to make open source software available to people everywhere. We believe the best way to fuel innovation is to give the innovators the technology they need.

About Canonical ›
Thank You!

ivan.dobos@canonical.com
Ubuntu