BUILDING OPENSHEIFT AND OPENSTACK PLATFORMS WITH RED HAT

Pilar Bravo, Senior Solution Architect, Red Hat
David Manchado, Infrastructure Architect, Produban
Alfredo Moralejo, Senior Domain Architect, Red Hat
Cristian Roldán, Middleware Architect, Produban
WHO WE ARE
WHO IS WHO

PILAR BRAVO
Senior Solution Architect
JBoss Middleware

ALFREDO MORALEJO
Senior Cloud Domain Architect

CRISTIAN ROLDAN
Middleware Architect

DAVID MANCHADO
Infrastructure Architect
A Global Company in 9 countries giving services to 120 Santander Group affiliates
SERVICES PROVIDED...

- 117 million Retail Banking Customers
- 11.6 million Online Banking Customers
- 30 million of credit cards
- 80 million of debit cards
- 30 million Contact Centre calls a month
- 1,258 million of weekly transactions
- 67 million of card transaction during peak days
- 2.4 million weekly batch executions
- 16.7 million of daily payments

ON TOP OF...

- 10 Corporate Datacenters
- 15 Mainframes
- + 28,000 physical servers
- + 64,000 logical servers
- + 22,000 data bases
- + 28,000 web app servers
- + 12,900 branches
- + 253,000 desktops
- + 6 PB/M data btw DC
GLOBAL CLOUD PROJECT

- Aims to provide a full XaaS stack
  - Already existing services
  - IaaS (NEW)
  - PaaS (NEW)

- Enable digital transformation (Banking 3.0)

- DevOps

- Mobile Apps
BUILDING AN OPENSTACK PLATFORM
DESIGN PRINCIPLES

Greenfield approach
General-purpose Cloud
Software Defined Everything
Multilocation
Scale-out
Failure domain contention
Vendor lock-in avoidance
Open Standards
OpenSource First (...but not only!)
DECISION MAKING PROCESS: OPENSTACK

WHY OPENSTACK
- Openness
- Community
- Interoperability
- Upgrade-in-place (starting from Icehouse)
- Technology meeting point (de-facto standard)

WHY RED HAT
- Close relationship since 2010
- Major player in OpenStack
- Professional Service offering
- Support
# DECISION MAKING PROCESS: SERVERS

<table>
<thead>
<tr>
<th>Openstack Services</th>
<th>Compute Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware</td>
<td>KVM</td>
</tr>
<tr>
<td>Traditional Standalone Server</td>
<td>OpenCompute</td>
</tr>
<tr>
<td>Local disk</td>
<td>Local disk (Ceph)</td>
</tr>
</tbody>
</table>

- Efficiency
- Data Center strategy
- Open

http://www.opencompute.org
DECISION MAKING PROCESS: STORAGE

Software Defined Storage
Multiple storage needs (image, block & object)
Scale-out
Openstack alignment
Maximum usage of available resources

OpenSource reference solution for OpenStack
Flexibility
Pay as you grow
Supported by Red Hat
...and it works!
DECISION MAKING PROCESS: NETWORKING

Software Defined Network
Non-proprietary fabric
Based on standard routing protocols (OSPF)
Leaf & Spine topology
Scalability
Openstack alignment
Avoid L2 adjacency

Federation Capabilities
Distributed routing
Maturity
Support
MULTILOCATION DEPLOYMENT

- Located on Corporate DataCenters
- Traditional failure domain approach
  1. Region
  2. Availability Zone (AZ)
- Provide building blocks to define resilient architectures on top
HIGH LEVEL DESIGN

Red Hat Enterprise Linux
OpenStack Platform

Horizon (Dashboard)

CEPH
CINDER
Block Store
SWIFT
Object Store
NEUTRON
Networks
GLANCE
Images
NOVA
Compute
HEAT
Orchestration
KEYSTONE
ID Management
CEILOMETER
Metering

Hypervisor

KVM

Hardware

lenovo
hp
huawei

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Red Hat CloudForms

by Red Hat Cloud

Red Hat CloudForms

OpenStack Platform

NEUTRON
Networks

NEUTRON
Networks

CEPH

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)

Public Cloud

Windows Azure
amazon
web services
SOFT LAYER

SATELLITE 6
(orchestration, automation and patch management)
SIZING

CURRENTLY

Biggest region: 88 compute nodes / 44 ceph OSD nodes (440 x 4TB OSDs)
Smallest region: 8 compute nodes / 8 ceph OSD nodes (80 x 4TB OSDs)
Total deployed: 160 compute nodes / 12 ceph OSD nodes (120 x 4TB OSDs)

MID TERM

14,000 CORES
200TB RAM
600TB OSD JOURNAL
16PB OSD CAPACITY

Think big, start small → plan to grow to ~ 1000 nodes
<table>
<thead>
<tr>
<th>Version</th>
<th>Private Cloud Instances</th>
<th>Block Storage</th>
<th>Software Defined Network</th>
<th>OS</th>
<th>Networking Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0</td>
<td>• RHEL Images (Ceph Volumes)</td>
<td>• Local Disk</td>
<td>• Private networks</td>
<td>RHEL 7.0</td>
<td>Nuage R3.03</td>
</tr>
<tr>
<td></td>
<td>• RHEL Images (Local Disk – Bigdata)</td>
<td>• Ceph volumes</td>
<td>• External GSNet network (floating IP, shared Subnet)</td>
<td>Openstack Icehouse</td>
<td>CEPH 1.2.2</td>
</tr>
<tr>
<td>v0.1</td>
<td>• RHEL Images (Ceph Volumes)</td>
<td>• Local Disk</td>
<td>• Service Chaining</td>
<td>RHEL 7.0</td>
<td>Nuage R3.06</td>
</tr>
<tr>
<td></td>
<td>• RHEL Images (Local Disk – Bigdata)</td>
<td>• Ceph volumes</td>
<td>• Network Templates</td>
<td>Openstack Icehouse</td>
<td>CEPH 1.2.2</td>
</tr>
<tr>
<td>v1.0 Beta</td>
<td>• RHEL Images (Ceph Volumes)</td>
<td>• Local Disk</td>
<td>• L3 Federation</td>
<td>RHEL 7.1</td>
<td>Nuage R3.06</td>
</tr>
<tr>
<td></td>
<td>• RHEL Images (Local Disk – Bigdata)</td>
<td>• Ceph volumes</td>
<td></td>
<td>Openstack Juno</td>
<td>CEPH 1.2.3</td>
</tr>
<tr>
<td>v1.0</td>
<td>• RHEL Images (Ceph Volumes)</td>
<td>• Local Disk</td>
<td></td>
<td>RHEL 7.1</td>
<td>Nuage R3.07</td>
</tr>
<tr>
<td></td>
<td>• RHEL Images (Local Disk – Bigdata)</td>
<td>• Ceph volumes</td>
<td></td>
<td>Openstack Juno</td>
<td>CEPH 1.2.3</td>
</tr>
<tr>
<td></td>
<td>• WIN Images (X Small to X Large)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Software Defined Network features include:
- Private networks
- External GSNet network (floating IP, shared Subnet)
- Service Chaining
- Network Templates
- L3 Federation
- External Internet Network
TECHNICAL CHALLENGES

• Think big, start small
• Maximize resource usage
• Non-cloud native workloads → Big Data
• Availability Zones isolation
• Live Architecture
• Heterogeneous components integration and lifecycle (HW, Openstack, SDS, SDN...)
• Non-openstack ecosystem integration (monitoring, billing, identity provider...)
DEPLOYMENT ARCHITECTURE

- Distribute control plane in following roles:
  - Load Balancers: haproxy
  - Backend: MariaDB, MongoDB, RabbitMQ
  - Controllers: OpenStack services

- Pacemaker as cluster manager
- Galera for MariaDB replication
- RabbitMQ with mirrored queues

- Additional per-AZ cluster with cinder
RESOURCE DISTRIBUTION

• Goal: maximize hardware resources usage
• Hyperconvergent mode not recommended by Red Hat.
• Approach: stability over performance
• Limit resources usage (specially memory) for ceph (OSDs) and nova (VMs):
  - cgroups to limit memory used by OSDs (~40GB)
  - Reserved_host_memory_mb to reduce the memory for nova scheduler (~50GB)
  - Use cinder QoS to limit per-volume resources
  - Distribution of available network bandwidth for different workflows (QoS)
CEPH DESIGN

3 Copies using a rule placing all copies in different racks and zones inside a given AZ/Region
THE DATA ANALYTICS CHALLENGE

- Critical use case: big data with hadoop and HDFS
  - Designed and conceived for bare metal with local disks
- Created several big flavors for analytics
- Main challenge: I/O access for HDFS

- Ironic
- PCI-Passthrough
- Cinder
- Ceph driver
- Swift
- Ephemeral
THE DATA ANALYTICS CHALLENGE (II)

- Defined non-converged nodes with local disks in a Host Aggregate
- Assigned extra_specs to analytics flavors to schedule in non-converged nodes
- At boot time, a libvirt hook attach virtual RAW disks on top of local disks to Vms
- Able to achieve required performance
OPENSTACK SEGREGATION

* Pending of fix for bz 1189887
OPENSTACK SEGREGATION (II)

- Independent CEPH cluster for each AZ for full isolation
- External replication script to clone images between ceph clusters
- Using glance multi-location to register all copies for each image
- Pending on patch in cinder to support CoW with multi-locations
- Next versions of cinder will allow glance to manage multiple RBD stores
NEXT STEPS

New OpenStack projects/features
- Trove
- Sahara
- Ironic
- Designate
- Manila
- LBaaS

Upgrading the whole installed base ¿twice a year/continuous?
Deploy pending regions / grow in the current ones
Object Storage (Swift-based)
Keystone integration with Identity Provider (SAML)
Cinder & QoS
Evolve architecture and fine tuning
BUILDING AN OPENSSHIFT PLATFORM
THE ENVIRONMENT

• Produban provides services to ISBAN

• ISBAN
  − Very focused on Websphere (own framework Banksphere)
  − Started migration of Banksphere to JBoss
  − Interest in:
    • JEE platform
    • Microservices approach
    • Self service for developers
    • ¿PaaS? ... sure!
PRODUBAN VS OPENSHIFT

Produban wanted to:

- Know what they were doing
- Understand the platform
- Be able to adapt the platform to their needs

Red Hat needed

- Defined requisites
- Set expectations and goals
- "Enable" Produban (as a partner)
INITIAL INSTALLATION

• First install was completely manual
• Installation guide became our “Book of knowledge”
• 3 people, 1 keyboard
  − (1 week of less than 2 hours keyboard time for consultant)
  − Required a lot of patience ... for all of us
INITIAL INSTALLATION OUTCOME

• Produban felt very comfortable with the product
• We needed a Solution, not a Product
  – Requisites were defined
  – Architecture was needed
  – Project roadmap needed
  – Platform not available
REQUISITES

- 45 infrastructure requisites defined
- 4 priority levels (from “Mandatory” to “Good to Have”)
  - Infrastructure
  - Operational
    - Upgrades were a very important topic
  - Backup
  - Monitoring
REQUISITES: GEARS

• Zones and Regions appeared with the perfect timing
• Gear sizes were used as Gear profiles permitting:
  – Allocate gears in DEV / PRE / PRO environments
  – Allocate gears in Europe or America region
  – Enable apps in Internet or Intranet
  – ... and of course, assign gear size
## Architecture: Regions, Zones, Districts

<table>
<thead>
<tr>
<th>Region</th>
<th>Zone Europe</th>
<th>Region America</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Zone Cantabria</td>
<td>Zone Mexico DF</td>
</tr>
<tr>
<td></td>
<td>Zone Madrid</td>
<td>Zone São Paulo</td>
</tr>
<tr>
<td>Districts / Gear Profiles</td>
<td>M M M</td>
<td>M M M</td>
</tr>
<tr>
<td>INT</td>
<td>L L L</td>
<td>L L L</td>
</tr>
<tr>
<td>PRO</td>
<td>M M M</td>
<td>M M M</td>
</tr>
<tr>
<td>NON-PRO</td>
<td>S S S</td>
<td>S S S</td>
</tr>
<tr>
<td>IT SMALL</td>
<td>M M</td>
<td>M M</td>
</tr>
<tr>
<td>IT LARGE</td>
<td>L L</td>
<td>L L</td>
</tr>
<tr>
<td>PRO</td>
<td>S S</td>
<td>S S</td>
</tr>
<tr>
<td>NON-PRO</td>
<td>M M</td>
<td>M M</td>
</tr>
<tr>
<td>IT SMALL</td>
<td>L L</td>
<td>L L</td>
</tr>
<tr>
<td>IT LARGE</td>
<td>L L</td>
<td>L L</td>
</tr>
</tbody>
</table>

#redhat #rhsummit
SOFTWARE CONFIG AND MANAGEMENT (I)

• Necessary

• Satellite 5 available (Satellite 6 in beta)
  – Used the corporate build to be in line with policies
  – Cloned Software Channels to keep a stable baseline
  – Created Config Channels for each role (Broker, Node, DB+Queue)
  – Created Activation Keys for each role
    • Associated Software Channels
    • Associated Config Channels
  – Support scripts for intermediate tasks
SOFTWARE CONFIG AND MANAGEMENT (II)

• Config channels kept versioned backup of configuration
  – Great to debug issues
  – Macros helpful for machine specific config
  – Customer loved “rhncfg-manager”
• New Nodes / Brokers / DB+Queue easily deployed
• No request for automatic deployment
  – Puppet considered for “phase 2” with Satellite 6
CUSTOM CARTRIDGES

- CA Wily Introscope
  - Created a cartridge to monitor apps:
    - JBoss
    - Tomcat

- Customer wanted to deploy plain Java apps
  - Created initially for Spring Boot applications.
    - Cartridge won the "Winter of Code"

https://github.com/Produban/ose_cartridge_javase
LOGGING

• OpenShift's Infrastructure
  – Centralized logging in place
  – Rsyslog for everything
  – Suggested ELK but not accepted (user permissions)

• Applications.
  – OSE's logshifter was tested, but found some performance issues.
  – Appender for Kafka is used.
MONITORING

• Centralized monitoring in place
  – Two levels of monitoring
    • OpenShift's Infrastructure
    • Applications
  – CA Wily Introscope
  – OpenShift Online scripts were used and improved

https://github.com/Produban/OpenShift20_Monitoring
OPENSШІFT INFRASTRUCTURE MONITORING
OPENS SHIFT OVERVIEW ON OPENNEBULA
OPENSSHIFT'S NODE MONITORING

OSE's metrics are generated by the command `oo-stats --format yaml`
OSE's metrics are generated by the command `oo-stats --format yaml`
OPENSHIFT'S BSN NODES MONITORING
OPENSHIFT CUSTOM LOADBALANCER MONITORING

OSS Project https://github.com/Produban/openshift-origin-app-load-balancer
CUSTOM LOAD BALANCER

- External load balancer not available
  - Let's make one!
  - Keepalived for floating IP
  - Nginx for redirection
  - Custom listener to manage queues
  - Mcollective for actions

https://github.com/Produban/openshift-origin-app-load-balancer

The custom Load Balancer is not used in Azure, multicast is not supported.
CONCLUSION (I)

• Produban is happy with OpenShift Enterprise 2.x
  – OSE is very flexible and open.
    • We love package oriented solutions instead of black box....
    • Easy to deploy in any IaaS.
  – We love cartridge specification.... much flexible than other PaaS solutions
  – Is not easy to achieve a stable OSE infrastructure.
  – Infrastructure custom monitoring solution is a MUST.
  – Intuitive and useful OpenShift's eclipse plugins.
  – ssh to GEAR is one of the most useful feature.
CONCLUSION (II)

• We have learned a lot of new things ...
  – Monolithic applications don't fit well in a PaaS environment.
  – PaaS is the perfect environment for Microservices applications.
  – The twelve-factor app, is the core pattern for PaaS applications
    http://12factor.net/build-release-run
  – PaaS administration team, why DevOps skill is a must ?
    • Installation, configuration and integration with external components is complex ...
    • Monitoring, lots of Ruby, Java, bash scripts ...
    • From development perspective PaaS is always the culprit ...
    • CI/CD/Maven/Git/Cartridge is a complex ecosystem for troubleshooting ...
PRODUBAN PAAS STRATEGY

PaaS Approach

Phase 1
- For DEV & Continuous integration
  - 1 region

Phase 2
- For DEV & Continuous integration
  - All regions

Phase 3
- For DEV & PROD
  - All regions

Service Catalog
OPENSHIFT 3 BETA

- We are involved in OpenShift 3 beta
  - Already tested OpenShift Origin Alpha.
  - Docker ecosystem is great!.
  - We have started with Drop 3.
  - Several teams were testing OpenShift V3 beta.
  - We have opened lots of issues in GitHub.

**IMPORTANT**

Service Marketplace: We feel very comfortable with Cloud Foundry Marketplace architecture, we would like to see something similar in OpenShift .... why not reuse the CF's Service Broker API?

http://docs.cloudfoundry.org/services/api.html
THE TEAM

PABLO  Pilar  Miguel Angel  Dani
Ania  Miguel  Cristian  Enrique
Alfredo  Sergio  Xavi  Pedro
Silvia  Raquel  Andrea  David
Miguel  David  Raul  Carlos
Carlos  Mark  Jonas  Oscar
Andrea  Juan  Miguel Angel  Roberto
Dani  Jose  Raquel  Javi
Nuria  Silvia  Xavi  Edu
Javier  lluis  Dani  Carlos
Antonio  Carlos  Manolo  Roberto

#redhat #rhsummit
Any questions?