Informatika v javni upravi 2018

"Slovenija – zelena referenčna država v digitalni Evropi"

10. in 11. december 2018 | Kongresni center Brdo pri Kranju

Building a government cloud

Concepts and Solutions

Dr. Gabor Szentivanyi, ULX Open Source Consulting & Distribution

Background

- Over 18 years of experience in enterprise grade open source
- Based in Budapest, Hungary
- Distinguished partner of Red Hat for Hungarian and regional business
- Consultancy in large scale datacenter, middleware and cloud design, implementation, integration, training
- Specialized in private and hybrid clouds based on leading Red Hat technology

ULX enterprise open source IT

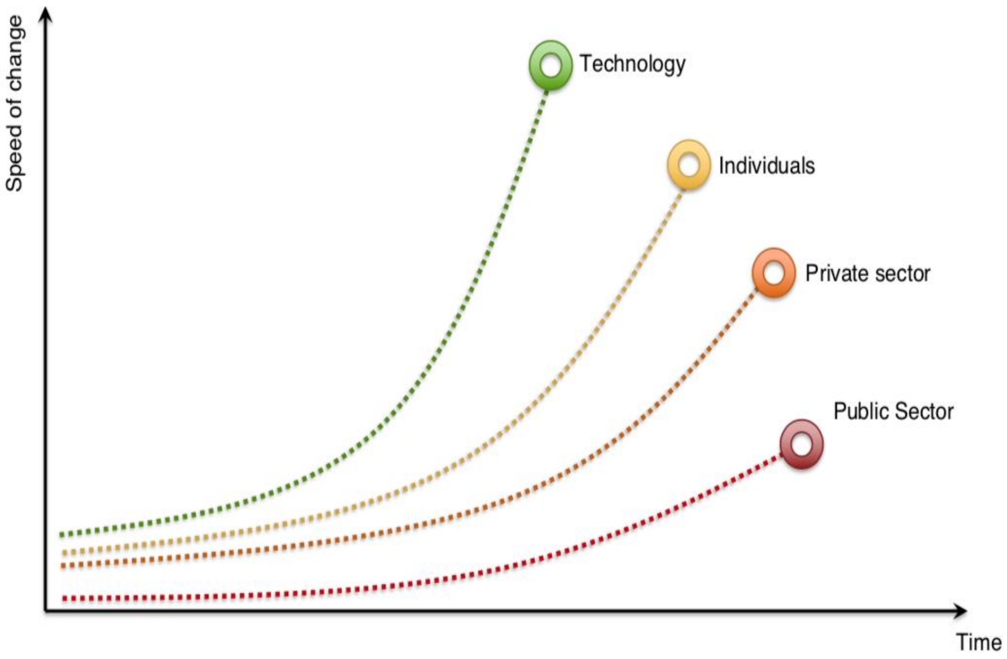
redhat

What’s in a cloud anyway

- Definition core: providing IT resources with elastic scalability, accountability
- What are these resources?
  - Low-level infrastructure services: computing, storage, network
  - Platform services for developers: container, DB, middleware, ….
  - Application services: standardized high-level components for complex application building
- Public vs. Private clouds
- Cloud provision vs. Cloud management
- Infrastructure vs. Platform vs. Application clouds
- Think big: Utilization, Entropy

Are government clouds different?

- There is a lot to do, perceptions are diverging
- Between private and public clouds
- Standardization vs. choice
- Requirement details blurry
- Outdated regulations
- Purchase process vs. agility
- Sizing, extension
- Chargeback
- Security

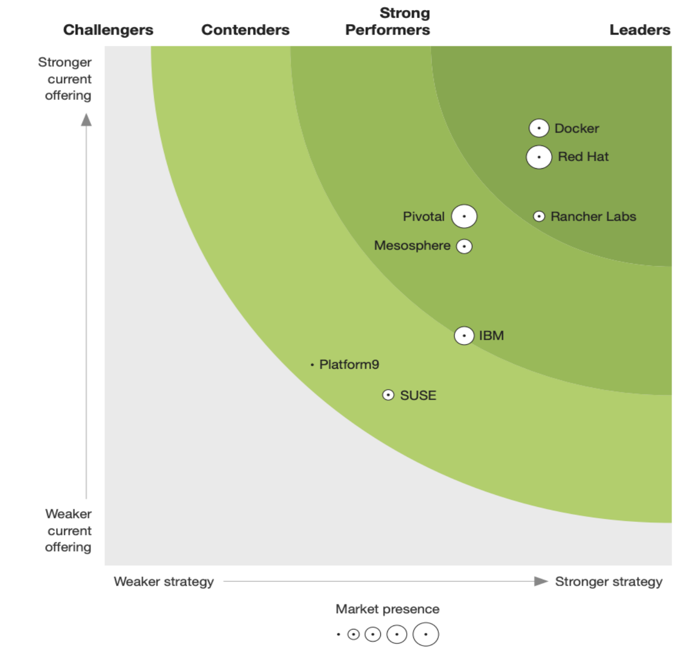


Source: Bersin et al (2017)
Deloitte University Press

The role of Red Hat in the cloud

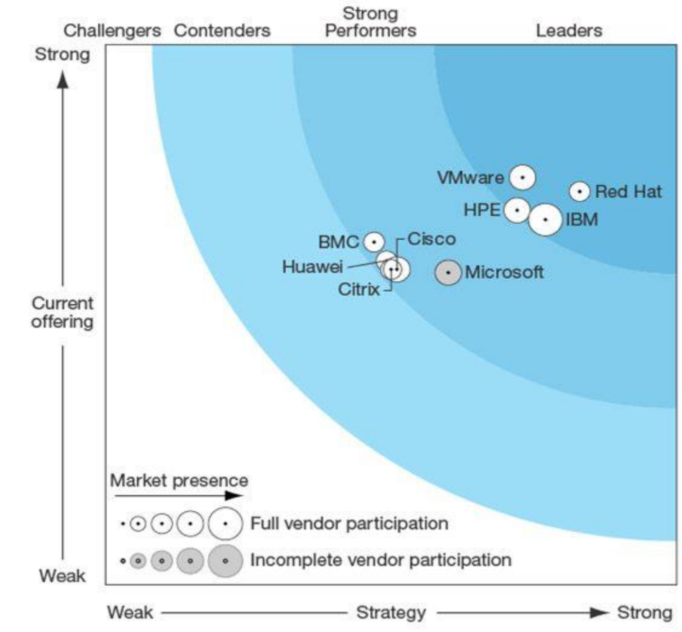
The cloud is open source: Linux, Java, Docker, OpenStack, Kubernetes, … but why?

THE FORRESTER NEW WAVE™
Enterprise Container Platform Software Suites
Q4 2018



What is the best tool in the market?

FIGURE 3 Forrester Wave™: Private Cloud Software Suites, Q1 '16



Most complete offering:

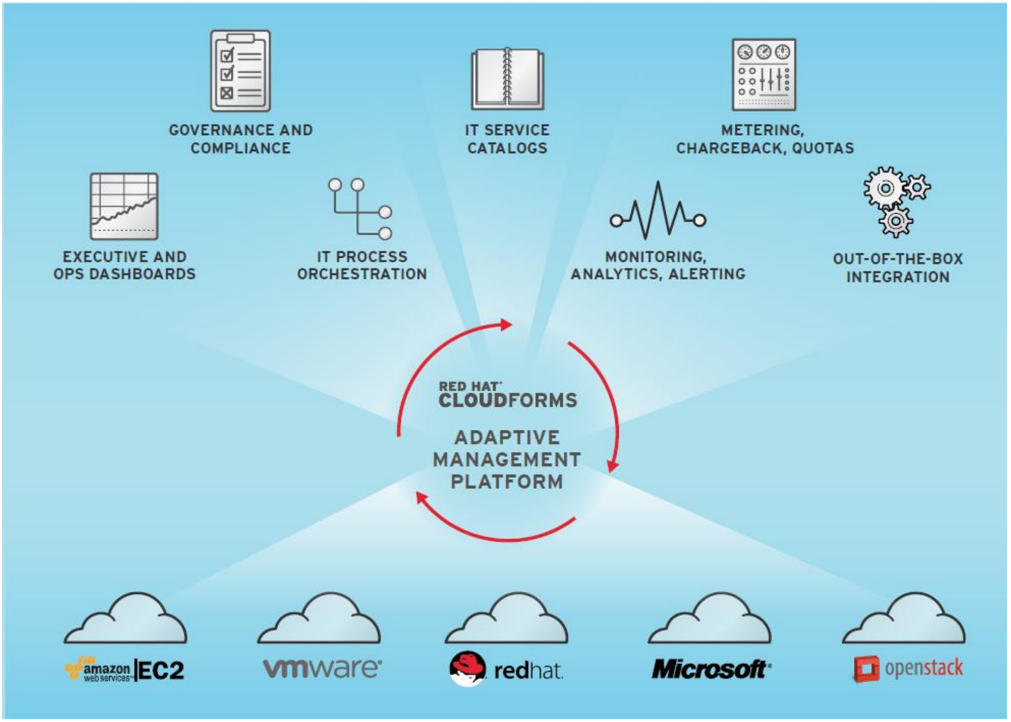
- Operating system
- Virtualization
- Infrastructure provider
- Platform and containers
- Cloud management framework

ULX enterprise open source IT

redhat

Red Hat CloudForms: What is included

- Powerful cloud management framework with overarching functionality
- Resource providers (private / public):
  - OpenStack, VMware, Hyper-V, RHV, OpenShift
  - Azure, Google Compute Engine, Amazon EC2


GOVERNANCE AND COMPLIANCE
IT SERVICE CATALOGS
METERING, CHARGEBACK, QUOTAS
EXECUTIVE AND OPS DASHBOARDS
IT PROCESS ORCHESTRATION
MONITORING, ANALYTICS, ALERTING
OUT-OF-THE-BOX INTEGRATION
RED HAT CLOUDFORMS
ADAPTIVE MANAGEMENT PLATFORM
amazon web services EC2
vmware
redhat
Microsoft
openstack

- GUI, Chargeback, Automation, Dashboards, Metering
- Service management (catalogs, lifecycle, bundles, …)
- Smart state analysis, policy / compliance

Project scope

Hardware: 6000 VMs, 3000 core, 135 TB RAM, 1200 TB storage

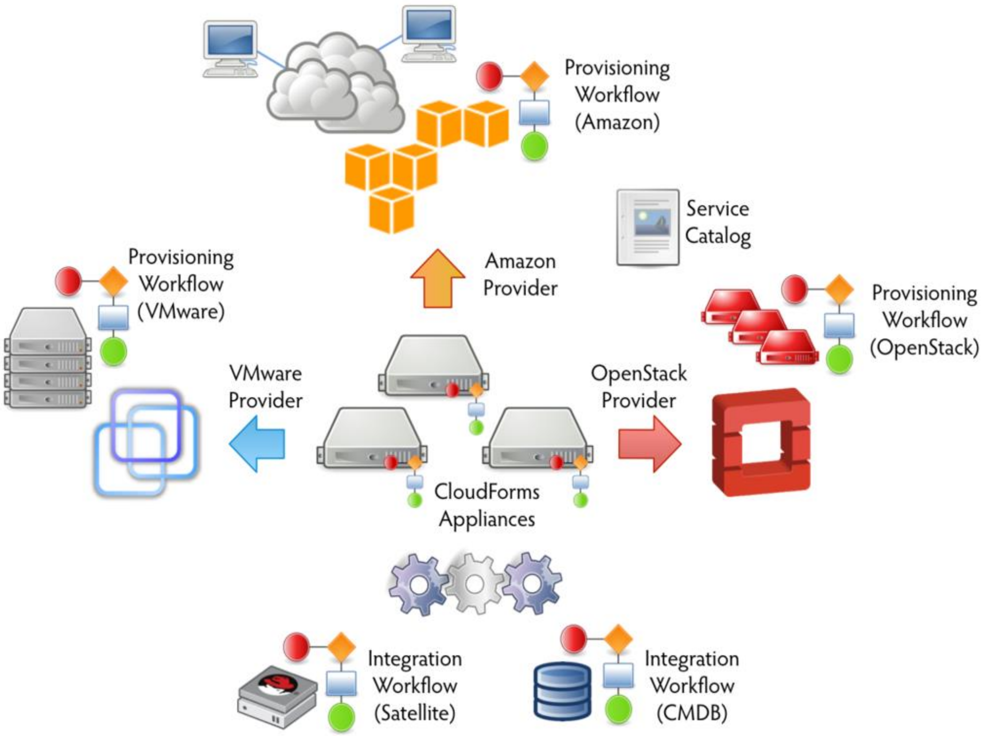
Cloud scope: infrastructure services with some platform level additions

Services designed, implemented, catalogized and provided for tenants

- Traditional virtualization: VM-based resources (VMware, RHV)
- Innovative cloud resources: containers (OpenShift), infrastructure (OpenStack)
- Application runtime environments (traditional)
- Automated network: IP, VLAN, DHCP, DNS, load balancer, firewall, VPN
- Database as a Service: Oracle, MS SQL, Postgres, MariaDB, MongoDB
- Centralized logging, monitoring, backup, virus check, snapshotting
- Centralized file services: NFS, CIFS
- Reporting, Policy and compliance enforcement
- Tenant provisioning and management, quota management, incident management

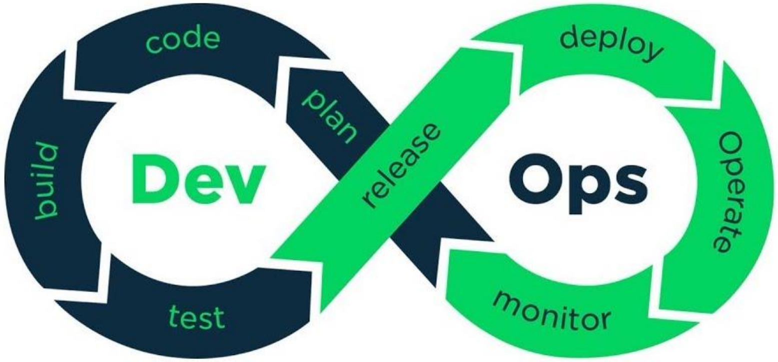
Strategic concepts and planning in general

- Plan thoroughly even if challenging, at least for 5-10 years
- Set expectations, and manage discrepancies and conflicts early
  - People especially decision makers easily expect all varieties
- Decouple from hardware lifecycles
- Start with large number of resources, think big!
- Define a delicate balance between standardization and freedom of choice
- Implement for today but plan for the future


Provisioning Workflow (Amazon)
Service Catalog
Provisioning Workflow (VMware)
Amazon Provider
Provisioning Workflow (OpenStack)
VMware Provider
OpenStack Provider
CloudForms Appliances
Integration Workflow (Satellite)
Integration Workflow (CMDB)

Strategic concepts and planning in particular

- Automate everything you can
  - use smart policies rather than mimic manual workflows
- Design a separate test environment (really separate)
- “Infrastructure as code”
  - Treat infrastructure as code: versions, revisions, bugfixing
  - Use version management
  - Deploy infrastructure as code
- Prepare automated testing
  - Mostly at service level
- Use an agile devops approach



Areas of attention: legacy systems

- Tectonic shift between legacy and cloud
  - but cloud management is not for cloud ready applications only, legacy systems will coexist (especially in government)
- Cloud native architecture is service and application oriented, underlying infrastructure can be easily standardized
- Legacy is heavily infrastructure dependent, no straightforward way to fit in
- What do we do with legacy systems
  - Migrate: take advantage of the cloud without disrupting operations
  - Integrate: let them easily communicate with cloud applications

Areas of attention: robustness and scalability

- Heterogeneous components and their capability limitations are a challenge
  - each component needs to be validated
- Disaster recovery
  - A chain is only as strong as its weakest link
- Multi tenancy
  - Most legacy systems are not prepared
- Scalability
  - Cloud native vs. legacy
- Replaceability
  - Achieve independence from or build abstraction for special HW / SW components

From the tenant’s viewpoint

- Who is a tenant?
- Onboarding (contractual, technical)
- Selection of services offered
- Service catalog: simple services, service bundles
- Service lifecycle management (create, modify, retire)
- Tenant administration
- Centralized network services
- Reporting (resources, network, usage, utilization etc.)

Lessons learned

- Project runtime between 8-16 months, depends on prior preps
- All components selected need to scale or be made scale
  - *If not, be prepared to replace components*
- All components selected need to be multi-tenant
  - *If not, add many extra months to implement*
- Must be hardware independent, it's not the decisive factor
  - *If not, be prepared for trouble when extending that system*
- Save budget and time for excessive Change Requests
  - *If not, you will not meet expectations, great chance to fail*
- Train users heavily
  - *If not, they will find the system inappropriate*

Thank you for your attention!

Questions?

Dr. Gabor Szentivanyi, CEO

ULX Open Source Consulting & Distribution

gabor.szentivanyi@ulx.hu