Overview: Building Open Source Cloud Computing Environments

LinuxCon Europe 2011
Prague, Czech Republic

Mark R. Hinkle
Director, Cloud Computing Community
Citrix Systems Inc.
Twitter: @mrhinkle
Email: mrhinkle@cloudstack.org
Agenda

1. Introduction
2. Quick Cloud Computing Overview
3. Open Source Building Blocks for Cloud Computing
4. Open Source Tools for Cloud Management
5. Questions
%whoami

- Responsible for Driving Adoption of CloudStack Open Source Cloud Computing Software
- Joined Citrix via Cloud.com acquisition July 2011
- Former manager of Zenoss Open Source project 100,000 users, 1.5 million downloads
- Former Linux Desktop Advocate (Zealot?)
- Former LinuxWorld Magazine Editor-in-Chief
- Open Management Consortium organizer
- Author - “Windows to Linux Business Desktop Migration” – Thomson
- NetDirector Project - Open Source Configuration Management Project
- Sometimes Author and Blogger at SocializedSoftware.com/NetworkWorld
Quick Cloud Computing
Overview: *Or the Obligatory “What is the Cloud?” Slides*
Five Characteristics of Clouds

1. On-Demand Self-Service
2. Broad Network Access
3. Resource Pooling
4. Rapid Elasticity
5. Measured Service
Cloud Computing Service Models

**USER CLOUD a.k.a. SOFTWARE AS A SERVICE**

Single application, multi-tenancy, network-based, one-to-many delivery of applications, all users have same access to features.

*Examples: Salesforce.com, Google Docs, Red Hat Network/RHEL*

**DEVELOPMENT CLOUD a.k.a. PLATFORM-AS-A-SERVICE**

Application developer model, Application deployed to an elastic service that autoscales, low administrative overhead. No concept of virtual machines or operating system. Code it and deploy it.

*Examples: Google AppEngine, Windows Azure, Rackspace Site, Red Hat Makara*

**SYSTEMS CLOUD a.k.a INFRASTRUCTURE-AS-A-SERVICE**

Servers and storage are made available in a scalable way over a network.

*Examples: EC2, Rackspace CloudFiles, OpenStack, CloudStack, Eucalyptus, Ubuntu Enterprise Cloud, OpenNebula*
Deployment Models
Public, Private & Hybrid Clouds
Cloud Still Requires Architectural Design

• Cloud Computing isn’t a magical solution

• Design your architecture with the end in mind

• Make your infrastructure easily replicable
Building Compute Clouds with Open Source Software
Why **Open Source?**

- User-Driven Solutions to Real Problems
- Lower barrier to participation
- Larger user base, users helping users
- Aggressive release cycles stay current with the state-of-the-art
- Open data, Open standards, Open APIs
Principles of Open Cloud

- **Interoperability** (the ability to exchange and use information) between cloud computing products and services is **required** for unfettered competition between vendors and unrestricted choice for users.

- Users **must** be able to come (no barriers to entry) and go (no barriers to exit) regardless of who they are (no discrimination) and what systems they use (technological neutrality).

- Supporting vendors **must** therefore cooperate on standards, implementing those that exist (where applicable) and collaborating via an open process to develop those that don't, with a view to competing fairly on quality.

[http://www.opencloudinitiative.org](http://www.opencloudinitiative.org)
Open Virtual Machine Formats

Open Virtualization Format (OVF) is an open standard for packaging and distributing virtual appliances or more generally software to be run in virtual machines.

Formats for hypervisors/cloud technologies:

- Amazon - AMI
- KVM – QCOW2
- VMware – VMDK
- Xen – IMG
Sourcing OSS VMs and Cloud Appliances

- BoxGrinder
- BitNami
- SUSE Studio
Open Source Hypervisors

Open Source
- Xen, Xen Cloud Platform (XCP)
- KVM – Kernel-based Virtualization
- VirtualBox* - Oracle supported Virtualization Solutions
- OpenVZ* - Container-based, Similar to Solaris Containers or BSD Zones
- LXC – User Space chrooted installs

Proprietary
- VMware
- Citrix XenServer
- Microsoft Hyper-V
- OracleVM (Based on OS Xen)
### Open Source Compute Clouds

<table>
<thead>
<tr>
<th>Year Started</th>
<th>License</th>
<th>Hypervisors Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>GPL</td>
<td>Xenserver, Xen Cloud Platform, KVM, VMware</td>
</tr>
<tr>
<td>2006</td>
<td>GPL</td>
<td>Xen, KVM, VMware</td>
</tr>
<tr>
<td>2010 (Developed by NASA by Anso Labs previously)</td>
<td>Apache</td>
<td>VMware ESX and ESXi, Microsoft Hyper-V, Xen, KVM and Virtual Box</td>
</tr>
</tbody>
</table>

Other open source compute software include Abiquo, Red Hat’s CloudForms and OpenNebula. Numerous companies are building cloud software on OpenStack including Nebula, Piston Inc.
## Open Source Platform-as-a-Service

<table>
<thead>
<tr>
<th>Year Started</th>
<th>Sponsors</th>
<th>Hypervisors Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>VMware</td>
<td>Spring, Rails, Node.js</td>
</tr>
<tr>
<td>2011</td>
<td>Joyent</td>
<td>Node.js</td>
</tr>
<tr>
<td>2011</td>
<td>Red Hat</td>
<td>Express – Ruby, Php and Python</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flex – Jboss, Java EE6</td>
</tr>
<tr>
<td>2010</td>
<td>WSO2</td>
<td>Jboss, Java EE6</td>
</tr>
</tbody>
</table>
Open Source Cloud Computing Storage

- **GlusterFS** – Scale Out NAS system aggregating storage over Ethernet or Infiniband
- **Ceph** – Distributed file storage system developed by DreamHost
- **OpenStack Object Storage (SWIFT)** – Long-term storage object storage system
- **Sheepdog** – Distributed storage for KVM hypervisors
- **NFS** – Old standby, tried and true, not designed for cloud scale or performance
Cloud APIs Aren’t Created Equal
Open Source Abstractions

- jclouds
- libcloud
- deltacloud
- fog
Managing Clouds with Open Source Tools
Automation Unlocks the Potential of the Cloud

- MeatCloud, Can’t Keep up with Cloud Computing
- DevOps & Agile IT Philosophy
- Script Repetitive Tasks
- Automate, Automate, Automate
### The Myth of the Nines

<table>
<thead>
<tr>
<th>Availability %</th>
<th>Downtime per Year</th>
<th>Downtime per Month</th>
<th>Downtime per Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>99.9% (three nines)</td>
<td>8.76 hours</td>
<td>43.2 minutes</td>
<td>10.1 minutes</td>
</tr>
<tr>
<td>99.95%</td>
<td>4.38 hours</td>
<td>21.56 minutes</td>
<td>5.04 minutes</td>
</tr>
<tr>
<td>99.99% (four nines)</td>
<td>52.6 minutes</td>
<td>4.32 minutes</td>
<td>1.01 minutes</td>
</tr>
<tr>
<td>99.999% (five nines)</td>
<td>5.26 minutes</td>
<td>25.9 seconds</td>
<td>6.05 seconds</td>
</tr>
<tr>
<td>99.9999% (six nines)</td>
<td>31.5 seconds</td>
<td>2.59 seconds</td>
<td>.0605 seconds</td>
</tr>
</tbody>
</table>

Average polling interval for monitoring - 5 minutes
Even superhuman operations people can't be alerted and take action in under 5 minutes.
One outage per year could drop service level to three nines or worse.
4 Types of Management Tools

**Provisioning**
Installation of operating systems and other software

**Configuration Management**
Sets the parameters for servers, can specify installation parameters

**Orchestration/Automation**
Automate tasks across systems

**Monitoring**
Records errors and health of IT infrastructure
Management Toolchains

Toolchain (n):
A set of tools where the output of one tool becomes the input of another tool.
## Open Source Provisioning Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Year Started</th>
<th>Language</th>
<th>License</th>
<th>Installation Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kickstart</td>
<td></td>
<td>Python</td>
<td>GPL</td>
<td>Most .dep and RPM based Linux distros</td>
</tr>
<tr>
<td><strong>Cobbler</strong> (Plus koan for PXE boot of VMs)</td>
<td>2007</td>
<td>Python</td>
<td>GPL</td>
<td>Red Hat, OpenSUSE, Fedora, Debian, Ubuntu</td>
</tr>
<tr>
<td>Spacewalk</td>
<td>2008</td>
<td>Perl, Python, Java</td>
<td>GPL</td>
<td>Fedora, Centos</td>
</tr>
<tr>
<td>Crowbar</td>
<td>2011</td>
<td>Ruby</td>
<td>Apache</td>
<td>(Bare metal provisioning)</td>
</tr>
<tr>
<td>Tool</td>
<td>Year Started</td>
<td>Language</td>
<td>License</td>
<td>Client/Server</td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td>----------</td>
<td>---------</td>
<td>---------------------</td>
</tr>
<tr>
<td>Cfengine</td>
<td>1993</td>
<td>C</td>
<td>Apache</td>
<td>Yes</td>
</tr>
<tr>
<td>Chef</td>
<td>2009</td>
<td>Ruby</td>
<td>Apache</td>
<td>Chef Solo – No Chef Server - Yes</td>
</tr>
<tr>
<td>Puppet</td>
<td>2004</td>
<td>Ruby</td>
<td>GPL</td>
<td>yes</td>
</tr>
</tbody>
</table>
## Open Source Monitoring Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>License</th>
<th>Type of Monitoring</th>
<th>Collection Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cacti</strong></td>
<td>GPL</td>
<td>Performance</td>
<td>SNMP, syslog</td>
</tr>
<tr>
<td><strong>RRDTool</strong></td>
<td>GPL</td>
<td>Performance</td>
<td>SNMP, syslog</td>
</tr>
<tr>
<td><strong>Nagios</strong></td>
<td>GPL</td>
<td>Availability</td>
<td>SNMP, TCP, ICMP, IPMI, syslog</td>
</tr>
<tr>
<td><strong>Zabbix</strong></td>
<td>GPL</td>
<td>Availability/Performance and more</td>
<td>SNMP, TCP/ICMP, IPMI, Synthetic Transactions</td>
</tr>
<tr>
<td><strong>Zenoss</strong></td>
<td>GPL</td>
<td>Availability, Performance, Event Management</td>
<td>SNMP, ICMP, SSH, syslog, WMI</td>
</tr>
</tbody>
</table>
# Open Source Automation/Orchestration Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Year Started</th>
<th>Language</th>
<th>License</th>
<th>Client/Server</th>
<th>Support Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capistrano</td>
<td>2006</td>
<td>Ruby</td>
<td>MIT</td>
<td>Yes</td>
<td>None</td>
</tr>
<tr>
<td>RunDeck</td>
<td>2010</td>
<td>Java</td>
<td>Apache</td>
<td>Yes</td>
<td>DTO Solutions</td>
</tr>
<tr>
<td>Func</td>
<td>2007</td>
<td>Python</td>
<td>GPL</td>
<td>Yes</td>
<td>Fedora Project</td>
</tr>
<tr>
<td>MCollective</td>
<td>2009</td>
<td>Ruby</td>
<td>Apache</td>
<td>Yes</td>
<td>PuppetLabs</td>
</tr>
</tbody>
</table>
Conceptual Automated Toolchain

BootStrapped Image
- CloudStack
- OpenStack
- Eucalyptus

Provision
- Cobbler
- Kickstart

Configuration
- Puppet
- Chef

Monitoring
- Nagios
- Zenoss
- Cacti

Start/Stop Services
- RunDeck
- Capistrano
- MCollective
Questions?

Slides Can be Viewed and Downloaded at:

http://www.slideshare.net/socializedsoftware/
Contact Me

Professional: mrhinkle@cloustack.org
Personal: mrhinkle@gmail.com

Professional: 919.228.8049

Professional: http://www.cloudstack.org
Personal: http://www.socializedsoftware.com

Twitter: @mrhinkle
Additional Resources

- Devops Toolchains Group
- DevOps Wikipedia Page
- Open Cloud Initiative
- NIST Cloud Computing Platform
- Open Virtualization Format Specs
- Clouderati Twitter Account
- Planet DevOps
Crash Course in Open Source Cloud Computing

by Mark R. Hinkle is licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License.