Hitchhiker's Guide to Open Source Cloud Computing

By Mark R. Hinkle
Senior Director, Cloud Computing Community
Citrix Inc.
Twitter: @mrhinkle
Blog: www.socializedsoftware.com
Email: mrhinkle@socializedsoftware.com
%whoami

- Dedicated to the success of the Apache CloudStack & Xen Open Source Cloud Communities at Citrix
- Conduct Build A Cloud learning activities all over the world
- Joined Citrix via Cloud.com acquisition July 2011
- Zenoss Open Source project to 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 Open Source Subnet
Why Open Source and the Cloud?

• User-Driven Context from Solving Real Problems
• Lower Barrier to Participation
• Larger user base, users helping users
• Aggressive release cycles stay current with the state-of-the-art
• Open Source innovating faster than commercial
• Open data, Open standards, Open APIs
Quick Cloud Computing Overview or the Obligatory “What is the Cloud Explanation”
Five Characteristics of Cloud

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: VMware CloudFoundry, Google AppEngine, Windows Azure, Rackspace Sites, Red Hat OpenShift, Active State Stackato, Appfog

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, OpenNebula
Deployment Models: Public, Private & Hybrid
Building Open Source Clouds
Cloud Architecture
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 (based
- Microsoft Hyper-V
- OracleVM (Based on OS Xen)
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
- VHD – Virtual Hard Disk - Hyper-V
## Sourcing Cloud Appliances

<table>
<thead>
<tr>
<th>Tool/Project</th>
<th>What you can do with them</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bitnami</strong></td>
<td>BitNami provides free, ready to run environments for your favorite open source web applications and frameworks, including Drupal, Joomla!, Wordpress, PHP, Rails, Django and many more.</td>
</tr>
<tr>
<td><strong>Boxgrinder</strong></td>
<td>BoxGrinder is a set of projects that help you grind out appliances for multiple virtualization and Cloud providers</td>
</tr>
<tr>
<td><strong>Oz</strong></td>
<td>Command-line tool that has the ability to create images for common Linux distributions to run on KVM</td>
</tr>
<tr>
<td><strong>SUSE Studio</strong></td>
<td>SUSE Studio supports building and deploying directly to cloud services such as Amazon EC2.</td>
</tr>
</tbody>
</table>
**Scale-Up or Scale-Out**

**Vertical Scaling (Scale-Up)** – Allocate additional resources to VMs, requires a reboot, no need for distributed app logic, single-point of OS failure

**Horizontal Scaling (Scale-Out)** – Application needs logic to work in distributed fashion (e.g. HA-Proxy and Apache, Hadoop)
## Compute Clouds (IaaS)

<table>
<thead>
<tr>
<th>Compute Clouds</th>
<th>Year Started</th>
<th>License</th>
<th>Virtualization Technologies</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apache CloudStack</strong></td>
<td>2008</td>
<td>Apache</td>
<td>Xenserver, Xen Cloud Platform, KVM, VMware (Hyper-V developing)</td>
</tr>
<tr>
<td><strong>Eucalyptus</strong></td>
<td>2006</td>
<td>GPL</td>
<td>Xen, KVM, <em>VMware</em> <em>(commercial version)</em></td>
</tr>
<tr>
<td><strong>OpenNebula</strong></td>
<td>2005</td>
<td>Apache</td>
<td>Xen, KVM, VMware</td>
</tr>
<tr>
<td><strong>OpenStack</strong></td>
<td>2010 (Developed by NASA by Anso Labs previously)</td>
<td>Apache</td>
<td>VMware ESX and ESXi, , Xen, Xen Cloud Platform, KVM, LXC, QEMU and Virtual Box</td>
</tr>
</tbody>
</table>

Numerous companies are building cloud software on OpenStack including Nebula, Piston Inc., CloudScaling.
OpenStack – Ecosystem of Projects

20 Collective projects hosted at:

https://launchpad.net/openstack
Cloud APIs

- jclouds
- libcloud
- deltacloud
- fog
# Cloud Computing Storage

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GlusterFS</strong></td>
<td>Scale Out NAS system aggregating storage over Ethernet or Infiniband</td>
</tr>
<tr>
<td><strong>Ceph</strong></td>
<td>Distributed file storage system developed by DreamHost</td>
</tr>
<tr>
<td><strong>OpenStack Storage</strong></td>
<td>Long-term object storage system</td>
</tr>
<tr>
<td><strong>Sheepdog</strong></td>
<td>Distributed storage for KVM hypervisors</td>
</tr>
</tbody>
</table>
## Platform-as-a-Service (PaaS)

<table>
<thead>
<tr>
<th>Project</th>
<th>Year Started</th>
<th>Sponsors</th>
<th>Languages/Frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CloudFoundry</strong></td>
<td>2011</td>
<td>VMware</td>
<td>Spring for Java, Ruby for Rails and Sinatra, node.js, Grails, Scala on Lift and more via partners (e.g. Python, PHP)</td>
</tr>
<tr>
<td><strong>Cloudify</strong></td>
<td>2012</td>
<td>Gigaspaces</td>
<td></td>
</tr>
<tr>
<td><strong>OpenShift</strong> **</td>
<td>2011</td>
<td>Red Hat</td>
<td>Java, Ruby, PHP, Perl and Python</td>
</tr>
<tr>
<td><strong>Stackato</strong></td>
<td>2012</td>
<td>ActiveState</td>
<td>Java, Python, PHP, Ruby, Perl, Node.js, others</td>
</tr>
<tr>
<td><strong>WSO2 Stratus</strong></td>
<td>2010</td>
<td>WSO2</td>
<td>Jboss, Java EE6</td>
</tr>
</tbody>
</table>
Software Defined Networking (SDN)
Overview of Software Defined Networking

Application Layer
- Business Applications

Control Layer
- SDN Control Software
- Network Services

Infrastructure Layer
- Network Devices

Control Data Plane Interface (e.g. OpenFlow)
## Why SDN?

<table>
<thead>
<tr>
<th>Cloud Promise</th>
<th>Cloud Reality</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Centralized Configuration and Automation</strong></td>
<td>Without true virtualization, network devices must still be manually configured.</td>
</tr>
<tr>
<td><strong>Instant Self-Service Provisioning</strong></td>
<td>In a physical network, it could take a long time for network engineer to provision new services.</td>
</tr>
<tr>
<td><strong>Elasticity and Scalability</strong></td>
<td>By horizontally scaling up the physical network, elasticity is lost.</td>
</tr>
<tr>
<td><strong>Designed for Failure</strong></td>
<td>Failover can be automated and physical network limitations can be alleviated.</td>
</tr>
</tbody>
</table>

Source: [Midokura](https://midokura.com)
OpenFlow enables networks to evolve, by giving a remote controller the power to modify the behavior of network devices, through a well-defined "forwarding instruction set". The growing OpenFlow ecosystem now includes routers, switches, virtual switches, and access points from a range of vendors.
## Software Defined Networking

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Floodlight</strong></td>
<td>The Floodlight controller is an enterprise-class, Apache-licensed, Java-based OpenFlow Controller.</td>
</tr>
<tr>
<td><strong>Indigo</strong></td>
<td>Indigo is an open source project to support OpenFlow on a range of physical switches. By leveraging hardware features of Ethernet switch ASICs, Indigo supports high rates for high port counts, up to 48 10-gigabit ports. Multiple gigabit platforms with 10-gigabit uplinks are also supported.</td>
</tr>
<tr>
<td><strong>OpenStack Networking “Quantum”</strong></td>
<td>Pluggable, scalable, API-driven network and IP management</td>
</tr>
<tr>
<td><strong>Open vSwitch</strong></td>
<td>Open vSwitch is a open source (ASL 2.0), multilayer virtual switch designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, SPAN, RSPAN, CLI, LACP, 802.1ag).</td>
</tr>
</tbody>
</table>
Big Data
1 Billion Facebook Users - October 2012

Source: Benphoster.com
Twitter at 400M Tweets Per Day – June 2012

Tweets in Millions

Source: TheBigDataGroup.com
Big Data and Storage Infrastructure

Data is growing faster than storage capacity and computing power. Legacy systems hold organizations back; storage software must include multi-petabyte capacity, support potentially billions of objects, and provide application performance awareness and agile provisioning.

-Gartner, Big Data Challenges for the IT Infrastructure Team
## Open Source NoSQL Databases

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Apache Cassandra</strong></td>
<td>Wide Column Store/ Families</td>
<td>API: many » Query Method: MapReduce, Replicaton: , Written in: Java, Concurrency: eventually consistent , Misc: like &quot;Big-Table on Amazon Dynamo alike&quot;, initiated by Facebook</td>
</tr>
<tr>
<td><strong>CouchDB</strong></td>
<td>Document Store</td>
<td>API: Memcached API+protocol (binary and ASCII) , most languages, Protocol: Memcached REST interface for cluster conf + management, Written in: C/C++ + Erlang (clustering), Replication: Peer to Peer, fully consistent, Misc: Transparent topology changes during operation, provides memcached-compatible caching buckets</td>
</tr>
<tr>
<td><strong>HBase</strong></td>
<td>Wide Column Store/ Families</td>
<td>API: Java / any writer, Protocol: any write call, Query Method: MapReduce Java / any exec, Replication: HDFS Replication, Written in: Java</td>
</tr>
<tr>
<td><strong>Redis</strong></td>
<td>Key Value/ Tuple Store</td>
<td>API: Tons of languages, Written in: C, Concurrency: in memory and saves asynchronous disk after a defined time. Append only mode available. Different kinds of fsync policies. Replication: Master / Slave, Misc: also lists, sets, sorted sets, hashes, queues.</td>
</tr>
<tr>
<td><strong>Riak</strong></td>
<td>Key Value / Tuple Store</td>
<td>API: JSON, Protocol: REST, Query Method: MapReduce term matching , Scaling: Multiple Masters; Written in: Erlang, Concurrency: eventually consistent (stronger then MVCC via Vector Clocks)</td>
</tr>
</tbody>
</table>
MapReduce

Problem Data

Solution Data

Master Node

Worker Node 1

Worker Node 2

Worker Node 3

Map

Reduce
Apache Hadoop

Overview
• Handles large amounts of data
• Stores data in native format
• Delivers linear scalability at low cost
• Resilient in case of infrastructure failures
• Transparent application scalability

Facts
• Apache top-level open source project
• One framework for storage and compute
  – HDFS – Scalable storage in Hadoop Distributed File System (HDFS)
  – Compute via the MapReduce distributed processing platform
• Domain Specific Language (DSL) - Java
Hadoop Architecture

**Non-Relational DB**
- **Hive**: Data warehouse that provides SQL interface. Ad hoc projection of data structure to unstructured.
- **HBase**: Column-oriented schema-less distributed DB modeled after Google's BigTable. Random real time read/write.

**Scripting**
- **Pig**: Platform for manipulating and analyzing large data sets. Scripting language for analysts.

**Machine Learning**
- **Mahout**: Machine learning libraries for recommendations, clustering, classifications and item sets.

**MapReduce**
- Parallel programming
- Handles large data blocks

**Hadoop Common**
- **HDFS**: Distributes & replicates data across machines
- **MapReduce**: Distributes & monitors tasks
Big Data Summary

• Quantity of Machine Created Data Increasing Drastically (examples: networked sensor data from mobile phones and GPS devices)
• Data manipulation moving from batched to real-time
• Cloud services giving everyone Big Data tools
• Consumer company speed and scale requirements driving efficiencies in Big Data storage and analytics
• New and broader number of data sources being meshed together
• Big Data Apps means using Big Data is faster and easier
Cloud Management Tools
Automation in the Cloud

Meat Cloud

Cloud Operations
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.
# Provisioning

<table>
<thead>
<tr>
<th>Project</th>
<th>Installation Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Axemblr Provisionr</strong></td>
<td>Can provision 10s to 1000s of machines on various clouds.</td>
</tr>
<tr>
<td><strong>Cobbler</strong></td>
<td>Distributed virtual infrastructure using koan (kickstart of a network to PXE boot VMs) for Red Hat, OpenSUSE Fedora, Debian, Ubuntu VMs</td>
</tr>
<tr>
<td><strong>JuJu</strong></td>
<td>Public Clouds - Amazon Web Services HP Cloud, Private OpenStack clouds, Bare Metal via MAAS.</td>
</tr>
<tr>
<td><strong>Salt Cloud</strong></td>
<td>Tool to provision “salted” VMs that can then be updated by a central server via ZeroMQ</td>
</tr>
<tr>
<td><strong>Crowbar</strong></td>
<td>(Bare metal provisioning)</td>
</tr>
</tbody>
</table>
# Configuration Management Tools

<table>
<thead>
<tr>
<th>Project</th>
<th>Year Started</th>
<th>Language</th>
<th>License</th>
<th>Client/Server</th>
</tr>
</thead>
<tbody>
<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 &amp; standalone</td>
</tr>
<tr>
<td>Salt</td>
<td>2011</td>
<td>Python</td>
<td>Apache</td>
<td>yes</td>
</tr>
</tbody>
</table>
# Automation/Orchestration Tools

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capistrano</strong></td>
<td>Utility and framework for executing commands in parallel on multiple remote machines, via SSH. It uses a simple DSL that allows you to define tasks, which may be applied to machines in certain roles</td>
</tr>
<tr>
<td><strong>RunDeck</strong></td>
<td>Rundeck is an open-source process automation and command orchestration tool with a web console.</td>
</tr>
<tr>
<td><strong>Func</strong></td>
<td>Func provides a two-way authenticated system for generically executing tasks, integrations with puppet and cobbler.</td>
</tr>
<tr>
<td><strong>MCollective</strong></td>
<td>The Marionette Collective AKA MCollective is a framework to build server orchestration or parallel job execution systems.</td>
</tr>
<tr>
<td><strong>Salt</strong></td>
<td>Execute arbitrary shell commands or choose from dozens of pre-built modules of common (or complex) commands.</td>
</tr>
<tr>
<td><strong>Scalr</strong></td>
<td>Provide scaling across multiple cloud computing platforms, integrates with Chef.</td>
</tr>
</tbody>
</table>
Conceptual Automated Toolchain

Generate Images
- SUSE Studio
- BoxGrinder

BootStrapped Image
- CloudStack
- OpenStack

Provision
- Cobbler
- SUSE Studio

Configuration
- Puppet
- Chef

Monitoring
- Nagios
- Zenoss
- Cacti

Start/Stop Services
- RunDeck
- Capistrano
- MCollective
NetFlix Open Source - ToolBag for AWS

http://netflix.github.com
Goodbye and thanks for all the fish!
Questions?

Slides Can be Viewed and Downloaded at: http://www.slideshare.net/socializedsoftware/

Copyright Mark R. Hinkle, available under the CCbySA license some rights reserved. 2012 -2013
Contact Me

Mark R. Hinkle
Senior Director,
Cloud Computing Community
Citrix Systems Inc.
Open Source Enthusiast

Professional: mark.hinkle@citrix.com
Personal: mrhinkle@gmail.com

Professional: 919.228.8049

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

Twitter: @mrhinkle
Appendix
Additional Resources

- Devops Toolchains Group
- Software Defined Networking: The New Norm for Networks (Whitepaper)
- DevOps Wikipedia Page
- NoSQL-Database.org – Ultimate Guide to the Non-Relational Universe
- Open Cloud Initiative
- NIST Cloud Computing Platform
- Open Virtualization Format Specs
- Clouderati Twitter Account
- Planet DevOps
- Nicira Whitepaper – It’s Time to Virtualize the Network
- Why Open vSwitch FAQ
## 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>Cacti / RRDTool</td>
<td>GPL</td>
<td>Performance</td>
<td>SNMP, syslog</td>
</tr>
<tr>
<td>Graphite</td>
<td>Apache 2.0</td>
<td>Performance</td>
<td>Agent</td>
</tr>
<tr>
<td>Nagios</td>
<td>GPL</td>
<td>Availability</td>
<td>SNMP, TCP, ICMP, IPMI, syslog</td>
</tr>
<tr>
<td>Zabbix</td>
<td>GPL</td>
<td>Availability/Performance and more</td>
<td>SNMP, TCP/ICMP, IPMI, Synthetic Transactions</td>
</tr>
<tr>
<td>Zenoss</td>
<td>GPL</td>
<td>Availability, Performance, Event Management</td>
<td>SNMP, ICMP, SSH, syslog, WMI</td>
</tr>
</tbody>
</table>