Server Management

Configure, Manage, Monitor

Linux Systems

Russell Doty
rdoty@redhat.com
Red Hat
Mid Level Sysadmin

Routine, day to day tasks

Experienced with Microsoft & VMware

New to Linux
Experienced Linux Sysadmin

Big Bag 'O Scripts & ssh

Powered by Google
OpenLMI Project Goals

- **Improve manageability of Linux systems**
  - Configuration of remote servers
  - Main Initial Focus: configure storage, networks, and authentication
  - Broader management support planned
- **Reduce learning curve for new SysAdmins to perform common Linux administration tasks**
- **Provide a foundation for advanced automation and interfaces**
- **Build on open industry standards**
- **Open and extensible**
  - Including 3rd parties and customers
What Can You Do With OpenLMI?

- Example: Storage
  - Determine what storage devices are connected to the system
    - Friendly device names & persistent device names
  - Partition, format, encrypt, create RAID sets, create logical volumes and volume groups, mount filesystems
    - Local & remote block devices
  - From programs, scripts or CLI
    - CLI example:
      "lmi> storage raid create –name R1 5 sdb sdc sdd sde sdf" (creates 5 disk RAID 5 set called R1)
    - More power and flexibility from LMIShell and OpenLMI API
  - Query, configure, monitor and change storage
What is OpenLMI?

- A set of Management Agents
  - That query, modify & monitor systems
- A Standardized Remote API
- A Standard Communications Infrastructure
  - Built on XML over https
- A Client Environment
  - CLI
  - Scripting (built on Python)
  - Other languages
OpenLMI Providers (Agents)

- Providers are Functional Modules
  - Get/Put attributes
  - Methods & relations
- Standard Interfaces
- Introspection
- Providers do all the work
- Toolchain for developing Providers
  - UML schema compile to produce skeleton
- Providers can be written in C/C++ or Python
- Providers can be call/response or asynchronous event driven
Events - OpenLMI Indications

• Notification of Asynchronous Events
  • e.g. job completion, drive failure, NIC changed state, password changed

• Usage:
  • Listener registers with CIMOM
    • Multiple Listeners per event
  • All registered Listeners notified when event occurs
  • Powerful tool for monitoring
OpenLMI Client Interfaces

- LMI
  - Enhanced CLI – interactive & BASH friendly
- LMIShell
  - Powerful Python scripting environment
  - Extensive library of Management Modules
  - Task Focused/Admin Friendly
  - Designed to be extended & modified
- REST – under development

Management System

- LMIShell Client
  - bash
  - LMI Commands
  - LMIShell Modules
  - Other Python Programs
  - Helper Functions
  - Native Python Objects
  - Python WBEM Interface

- C/C++ API
  - Powerful interface for writing Apps or integrating with existing Apps

- Java
  - Write Java Apps
  - Easy interface with JBoss
Manage Multiple Servers
Server Discovery

broadcast:

Who has LMI?

Who has LMI & SSSD?
Build Multiple Client Applications

Management System

Text Console

Graphical Console

Inventory System & CMDB
Java

Storage Expert System
JBoss & DROOLS

Disk Failed (event)
OpenLMI in RHEL 7 & Fedora 20

**Implementation:**
- DMTF/CIM technology stack
- HTTPS transport (no general Web server)

**Included Providers:**
- Storage
- Network
- System Services
- Power Management
- Local User Management
- Software Management
- System Monitoring (basic)
- System Information & Configuration
Sounds Interesting...
Show Me What You Are Talking About
Build a RAID Set

lmi> storage raid create --name myraid 5 vdb vdc vdd

lmi> storage vg create myvg myraid

lmi> storage lv create myvg myvol 1G

lmi> storage fs create xfs myvol

lmi> storage mount create \
    /dev/mapper/myvg-myvol \
    /mnt/myvol
Configure a NIC

lmi> net device list

<table>
<thead>
<tr>
<th>ElementName</th>
<th>OperatingStatus</th>
<th>MAC Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>em1</td>
<td>InService</td>
<td>3C:97:0E:4B:2E:53</td>
</tr>
<tr>
<td>lo</td>
<td>NotAvailable</td>
<td>00:00:00:00:00:00</td>
</tr>
<tr>
<td>virbr0-nic</td>
<td>NotAvailable</td>
<td>52:54:00:DF:BD:C4</td>
</tr>
<tr>
<td>wlp3s0</td>
<td>InService</td>
<td>60:67:20:C9:0B:DC</td>
</tr>
</tbody>
</table>

lmi> net address replace em1 192.168.0.130 24

lmi> net activate em1
Install an Application

lmi> sw show pkg httpd

    Name=httpd
    Arch=x86_64
    Version=2.4.9
    Release=2.fc20
    Summary=Apache HTTP Server
    Installed=no
    Description=The Apache HTTP Server is a powerful, efficient, and extensible web server.

lmi> sw install httpd

lmi> sw update httpd
Check if a Service is Running and Restart

```
  lmi> service show cups
       Name=cups
       Caption=CUPS Printing Service
       Enabled=Yes
       Status=Running

  lmi> service stop cups
       Stopped service "cups.service".

  lmi> service restart cups
       Restarted service "cups.service".
```
Reboot a Server

```lmi> power list

    PowerState       Available
suspend           yes
force reboot      yes
hibernate         yes
force poweroff    yes
poweroff          yes
reboot            yes

lmi> power reboot

lmi> power off```
The Elephant in the Room...
What About Puppet?

- Puppet and OpenLMI do different things
  - Puppet *puts system into a defined state*
  - OpenLMI *interacts with system*
- Puppet and OpenLMI are complementary
- Goal: for Puppet to be able to use OpenLMI to operate on systems
OpenLMI Security

- Communications encrypted using TLS (https)
  - Recommend using domain management such as Red Hat Identity Management (IdM)
- Authentication (Username & Password) required for all OpenLMI management operations
  - Authorization performed using standard admin accounts
- Works with SELinux in enforcing mode
- Planned future enhancement:
  - GSSAPI for domain based authentication
  - RBAC
Key Information

- Software: RHEL 7 or Fedora 20
- Project Website: www.openlmi.org
- Blog: techponder.wordpress.com
- Mailing list: lists.fedorahosted.org/mailman/listinfo/openlmi-devel
- IRC: #openlmi on freenode
Next Steps
Getting Started with OpenLMI

- Install OpenLMI on Fedora 20 or RHEL 7
- Easiest way is to use OpenLMI is with LMI command
- Study the existing LMISHell modules
  - Find something close to what you want to do
  - Hack up the LMISHell module
  - Consider submitting it to the upstream community
- Study the OpenLMI API documentation
- Use implementation language of your choice
  - Python, Java, C/C++,
    - REST or Ruby? Talk to us.
What You Can Do With OpenLMI - In Detail
What can you do with OpenLMI?

- Storage
- Networks
- Users
- Software
- Power
- System Services
- System configuration
- Active Directory
Storage Management

- All Storage
  - Partition
  - Format
  - RAID
  - Volume Management (Logical Volumes & Volume Groups)
  - Encryption
  - Mount
  - Query
  - Basic monitoring

- Local Storage
  - Enumerate Devices (drives)
  - Support for both persistent and friendly device names

- Remote Storage
  - Works today with any remote storage that can provide a LUN
  - Working with libstoragemgmt team to manage remote storage
Networking

- Goal: support all server features of Network manager
  - Enumerate and configure NICs
  - Configure network
  - Bonding
  - Bridging
Other Core Providers

- **Software**
  - List, install, remove, update rpm based software

- **Users**
  - Add, remove, edit local users

- **System Services**
  - List, state, start/stop/restart, modify settings
  - Add/Remove through Software Provider

- **Power**
  - Query and modify basic power states
  - Shutdown & reboot system
  - Need external support for Boot (e.g. BMC)
Other Core Providers (cont)

- Realmd
  - Integration with Active Directory for user authentication
  - Integration with Red Hat Identity Manager for user authentication
  - Configuration of Linux system to join domain
- Capabilities:
  - Query domain membership
  - Join a domain
  - Leave a domain
Other Providers (Not in RHEL 7.0)

- **Journald**
  - Read journald log records
  - Traverse log records using persistent iterators
  - Receive notification of new log entries
  - Write new log entries

- **SSSD**
  - Configure SSSD (System Services Security Daemon)

- **Firewalld**
  - Start/stop/restart firewall
  - Change firewall rules
  - Provides mechanism for central management of firewall

- **SELinux**
  - Query & set SELinux state
  - Manage booleans, ports & file labels
Other Providers (Not in RHEL 7.0)

- Performance Co-Pilot
  - List and select performance metrics
  - Reports a variety system statistics monitored by Performance Co-Pilot services.

- JON proof of concept
  - OpenLMI Providers function as JON Agents
  - Don't need JON/JBoss infrastructure on managed system
Key Technologies for OpenLMI Development

- Agent definition through modeling language – MOF files. UML based, defines the object model

- Toolchain to develop Agents
  - Generate code skeletons directly from MOF files
    - “C” - KonkretCMPI
    - Python – YAWN
  - LMI & LMIShell – powerful, friendly client environment & interface
    - Higher level of abstraction
    - Command expansion, integrated documentation, etc.
- Samples, examples, and actual agents
- Documentation and tutorials
Thank You!