Automated Testing of oVirt Node

Nov 09 2012

Fabian Deutsch <fabiand@redhat.com>
Software Engineer
Red Hat, Inc.
Agenda

- Introduction
- Installation and Configuration
- Test Automation
- Igor
- Current State & Future
Introduction
What is oVirt Node?

- Dedicated hypervisor
- Built on Fedora
- Firmware like
  - Install and forget about it
  - Limited set of functionality
  - TUI interface
Overview
Overview

• Packages from
  • Fedora
  • ovirt-node + ovirt-node-is

• ISO Image
  • Build using livecd-tools
  • (minimized) Size: ~170MB
  • Technologies: KVM, libvirt, vdsdm, …
Deployment

- Supported Media: CD/DVD, Flash, PXE

- Manual Installation
  - Using a newt based TUI
  - Storage

- Automatic Installation
  - Configuration with boot parameters
  - Storage, Network, SNMP, ...
Installation and Configuration

- Image is copied to LVM LV
  - Update: Boot the new image

- Basic configuration through TUI
  - Networking, oVirt Engine, SNMP, Kdump, …
  - Advanced configuration and monitoring through oVirt Engine

- All changes are persisted on /config
Test Automation
What needs testing?

- Deployment
  - Automate provisioning (real + virt)
- Installation & Updates
  - Different kernel arguments
  - Cover TUI installation
- Configuration
  - Handle temporary network loss
  - Cover TUI configuration
Automate: Deployment and Installation

- Installation via PXE
  - Import ISO using livecd-to-pxeboot
  - Create profile and system in Cobbler
    - Different kernel arguments for installation, update, reinstallation, ...
  - Run installation or update

- Works for virtual guests and real hardware
  - Libvirt for virtual guests
Igor

- Igord for client+profile life-cycle management
  - Concepts: Profile, Host, Testsuite

- oVirt Node Igor Plugin to run testcases on client
  - Handles interaction with TUI
## Jobs

<table>
<thead>
<tr>
<th>Job</th>
<th>State</th>
<th>Created at</th>
<th>Host</th>
<th>Profile</th>
<th>Testsuite</th>
</tr>
</thead>
<tbody>
<tr>
<td>IB6j38zRC2</td>
<td>running</td>
<td>Fr 20 Jul 2012 16:17:14 CEST</td>
<td>l-default-IB6j38zRC2</td>
<td>ovirt-node-iso-2.5.0-1.0.fc17.iso.edited.iso</td>
<td>al_extended</td>
</tr>
<tr>
<td>l6Tg38zRC2</td>
<td>failed</td>
<td>Fr 20 Jul 2012 16:09:24 CEST</td>
<td>l-default-l6Tg38zRC2</td>
<td>ovirt-node-iso-2.5.0-1.0.fc17.iso.edited.iso</td>
<td>mi_extended</td>
</tr>
<tr>
<td>IT2f38zRC2</td>
<td>passed</td>
<td>Fr 20 Jul 2012 16:02:53 CEST</td>
<td>l-default-IT2f38zRC2</td>
<td>ovirt-node-iso-2.5.0-1.0.fc17.iso.edited.iso</td>
<td>al_extended</td>
</tr>
<tr>
<td>iod238zRC2</td>
<td>passed</td>
<td>Fr 20 Jul 2012 15:56:35 CEST</td>
<td>l-default-iod238zRC2</td>
<td>ovirt-node-iso-2.5.0-1.0.fc17.iso.edited.iso</td>
<td></td>
</tr>
</tbody>
</table>
# Testsuites

<table>
<thead>
<tr>
<th>Testsuites</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>al_basic</td>
<td>Automated installation with minimal additional (no TUI) tests.</td>
</tr>
<tr>
<td>al_extended</td>
<td>Automated installation with many additional (including TUI) tests.</td>
</tr>
<tr>
<td>after_auto_install</td>
<td></td>
</tr>
<tr>
<td>installation_completed.sh</td>
<td>Source</td>
</tr>
<tr>
<td>collect_logs.sh</td>
<td>Source</td>
</tr>
<tr>
<td>set_admin_password.py</td>
<td>Source</td>
</tr>
<tr>
<td>basic</td>
<td></td>
</tr>
<tr>
<td>reboot</td>
<td></td>
</tr>
<tr>
<td>services</td>
<td></td>
</tr>
<tr>
<td>python</td>
<td></td>
</tr>
<tr>
<td>login</td>
<td></td>
</tr>
<tr>
<td>check_navigation</td>
<td></td>
</tr>
<tr>
<td>after_testing</td>
<td></td>
</tr>
<tr>
<td>ml_basic</td>
<td>Manual TUI installation with no additional tests.</td>
</tr>
<tr>
<td>ml_extended</td>
<td>Manual TUI installation with additional tests.</td>
</tr>
</tbody>
</table>

# Profiles

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>node-edited</td>
</tr>
<tr>
<td>ovirt-node-iso-2.5.0-1.0.fc17.iso.edited.iso</td>
</tr>
</tbody>
</table>

# Hosts

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ahost</td>
<td></td>
</tr>
<tr>
<td>bhost</td>
<td></td>
</tr>
</tbody>
</table>
Automate: Configuration

- Completely TUI based
Automate: Configuration

• Completely TUI based

• “Storyboard” based testing
  • Input through uinput
    - Create input devices in userspace (python-uinput)
    - Addresses console as well as TUI
    - Close to real input
  • Compare with screen via /dev/vcs
    - ... or any other python code
Current State

- Integrated into gerrit-jenkins workflow
  - junit-like reporting, visualized by Jenkins
- Works with virtual guests and real hardware
  - Currently run internally at Red Hat, results publicly available
- Basic testsuites (Summarized in a testplan)
  - Automated and TUI installation
  - TUI configuration
  - Some (sourcecode) sanity checks
Future

- Enhance testsuites
  - Different network setups
  - Testcases
    - cover all TUI pages in detail
    - for updates
    - for interaction with oVirt Engine

- Enhance Igor
Ressources

• oVirt Node
  • http://jenkins.ovirt.org
  • ML: node-devel@ovirt.org
  • ML: node-patches@ovirt.org
  • IRC: #ovirt on oftc.net

• Upstream
  • Jenkins http://jenkins-ci.org/
  • Gerrit http://code.google.com/p/gerrit/
  • Igor https://gitorious.org/ovirt/igord
Testcases

- A testcase is a script
  - Simple protocol: 0 is success
  - stderr/-out are logged
  - Annotate & Attach
- Communication with server between testcases
  - Network loss during a testcase is expected
- Server-side tracking of the testsuite state
  - Monitor timeout
Story

```python
story = [
    # Enter Nothing, wait 0 seconds, expect "Please Login" on screen
    (None, 0, "Please login"), # Ignore this for now.

    # Enter ..., wait ... seconds, expect ... on screen
    ("admin\n", 2, "Password:"),

    # Password (taken from set admin password)
    ("ovirt\n", 5, "Networking:")
]

if __name__ == "__main__":
    common.input.Storyboard("TUI login", story).run_and_exit()
```
Sourcecode

• Heavily relying on
  • Runtime informations
  • Config files

• Refactoring to enable testing
  • pyflakes, pylint, doctests
  • pytest
Jenkins

- Continuous Integration

- Jobs
  - To build rpms
  - To build the image

- Triggered by Gerrit (code review tool)
Igord

- Assign $profile to $host and run $testsuite
- Speaks to
  - Cobbler (Hosts, Profiles)
  - Libvirt (Hosts)
  - Filesystem (Hosts, Testsuites)
- Prepares disk images
  - LVM, Partitions, Filesystems
Autotest

• ... is an advanced testing framework

• TUI based testing is virt-only
  • Screenshot based

• Provisioning of machines is out of autotests scope
Foreman

- Already in sight
- Needs support in igord
- Until lately it wasn't possible to control the power of hosts
  - Now it is possible
Plugins

- RPM packages
- Existing image can be edited using edit-node
  - New image+plugin is composed