Aeolus

www.aeolusproject.org

An open source community-based cloud management software

Richard Su
rwsu@redhat.com
Red Hat, Inc
What is it?

• Not a Hypervisor or a Cloud
• It is a set of tools and applications that is used to build a portable cloud infrastructure
• Works with Amazon EC2, RHEV, VMWare vSphere, and more under development
Some Use Cases

- **Compute Farm**
  - Increase throughput by offloading work to a public cloud

- **Seasonality and Cost Savings**
  - Quickly bring up resources when needed
  - Pay for resources only when used
An Abstraction Layer

- Don't need to know each cloud provider's API
- Aeolus provides a way to
  - Create images that can be deployed to multiple cloud providers
  - Manage VMs across multiple cloud providers
Unlocks your cloud environment from any particular cloud vendor
Installation

- For Fedora, add aeolus repo to yum
- Install “aeolus-all” meta-package
- Run aeolus-configure. It uses puppet to configure the server and to bring up services and applications.
Account: mock

Settings

Provider Name: mock

Cloud type: Mock
Account name: mock

Username: mockuser
Password: ************
Priority: 
Quota: unlimited (Unlimited Quota)

Save  Delete Account
Images Can Be Built For

- Red Hat Enterprise Linux
- Fedora
- OpenSuse
- Debian
- Ubuntu
- Windows
- ScientificLinux
- and more

http://aeolusproject.org/oz.html
<template>
  <name>tmpl1</name>
  <description>foo</description>
  <os>
    <name>Fedora</name>
    <version>15</version>
    <arch>x86_64</arch>
    <install type="url">
      <url>http://download.fedoraproject.org/pub/fedora/linux/releases/15/Fedora/x86_64/os/</url>
    </install>
    <rootpw>changeme</rootpw>
  </os>
  <packages>
    <package name='aeolus-all'/>
  </packages>
  <repositories>
    <repository name="custom">
      <url>http://repos.fedorapeople.org/repos/aeolus/conductor/testing/fedora-15/x86_64/</url>
      <signed>false</signed>
    </repository>
  </repositories>
</template>
Choose one of the following options to upload or import an image into "default Environment".

**Upload**

**From URL**

- Name: fedora 15 image
- Choose a template file: `/home/rwsu/template.xml`
- Edit this file before saving: [Check box]

**Buttons:**
- Cancel
- Continue
<template>
  <name>fedora 15 image</name>
  <description>fog</description>
  <os>
    <name>Fedora</name>
    <arch>x86_64</arch>
    <version>15</version>
    <install type="url">
      <url>http://download.fedoraproject.org/pub/fedora/linux/releases/15/Fedora/x86_64/os/</url>
    </install>
    <rootpw>change-me</rootpw>
  </os>
  <packages>
    <package name="aeolus-all"/>
  </packages>
  <repositories>
    <repository name="custom">
      <url>http://repos.fedorapeople.org/repos/aeolus/conductor/testing/fedora-15/x86_64/</url>
    </repository>
  </repositories>
</template>
Valid Image

The markup and image references in "fedora 15 image" were successfully validated. You may now save the image, and create a deployable.

- Automatically make "fedora 15 image" deployable.
Create Deployable

Creating a deployable from the "fedora 15 image" component outline will allow it to be built and launched. Simply choose a catalog in which to store the deployable, and define a default cloud resource profile.

- Name: fedora 15 image
- Default Hardware Profile: hwp1
- Catalog: Performance Testing

[Save]
Performance Testing

Properties

Name: Performance Testing
Pool: Performance Eng

Deployables

<table>
<thead>
<tr>
<th>Name</th>
<th>Deployable XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>fedora 15 image</td>
<td>fedora 15 image</td>
</tr>
</tbody>
</table>
### Images

<table>
<thead>
<tr>
<th>Name</th>
<th>Images</th>
<th>HW Profile</th>
<th>HDD</th>
<th>RAM</th>
<th>ARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>fedora-15-image</td>
<td>1</td>
<td>hwp1</td>
<td>512</td>
<td>x86_64</td>
<td></td>
</tr>
</tbody>
</table>

### Build

Allow this deployable's associated image(s) to be built and pushed to cloud resource providers by enabling them below.

- **vsphere**
  - vsphere | vsphere
  - Images are not Built

- **rhevm**
  - rhevm | rhevm
  - Images are not Built

- **ec2**
  - ec2-us-east-1 | ec2
  - Images are not Built

- **mock**
  - mock | mock
  - Images are not Built
<table>
<thead>
<tr>
<th>Name</th>
<th>Images</th>
<th>HW Profile</th>
<th>HDD</th>
<th>RAM</th>
<th>ARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>fedora-15-image</td>
<td>1</td>
<td>hwp1</td>
<td>512</td>
<td>x86_64</td>
<td></td>
</tr>
</tbody>
</table>

Allow this deployable's associated image(s) to be built and pushed to cloud resource providers by enabling them below.

- **vSphere**
  - vsphere | vsphere
  - Some of the images are not pushed

- **RHEVM**
  - rhevm | rhevm
  - Some of the images are not pushed

- **EC2**
  - ec2-us-east-1 | ec2
  - Some of the images are not pushed

- **Mock**
  - mock | mock
  - Some of the images are not pushed
**fedora-15-image**

### Images

<table>
<thead>
<tr>
<th>Name</th>
<th>Images</th>
<th>HW Profile</th>
<th>HDD</th>
<th>RAM</th>
<th>ARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>fedora-15-image</td>
<td>1</td>
<td>hwp1</td>
<td>512</td>
<td></td>
<td>x86_64</td>
</tr>
</tbody>
</table>

**Images Valid**

### Build

Allow this deployable's associated image(s) to be built and pushed to cloud resource providers by enabling them below.

#### vsphere

- vsphere | vsphere  
- Images are being pushed.  

#### rhevm

- rhevm | rhevm  
- Images are being pushed.  

#### ec2

- ec2-us-east-1 | ec2  
- All Images are pushed and recent.  

#### mock

- mock | mock  
- All Images are pushed and recent.  

---

**Presented by Red Hat**
New Deployment to Performance eng Pool

Deployment Details

Deployment Name (uniquely identifies your deployment)

deploy1

Name available

Deployable

fedora-15-image

Realm

ec2-us-east-1

Auto-select

rhevm

vsphere

ec2-us-east-1

mock

PRESENTED BY RED HAT
Deltacloud

- Cloud abstraction API
- REST based, language independent
- Supports most major infrastructure as a service clouds
<table>
<thead>
<tr>
<th>Service</th>
<th>Create new instances</th>
<th>Start stopped instances</th>
<th>Stop running instances</th>
<th>Reboot running instances</th>
<th>Destroy instances</th>
<th>List all/get details about hardware profiles</th>
<th>List all/get details about realms</th>
<th>List all/get details about images</th>
<th>List all/get details about instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon EC2</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Eucalyptus</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>IBM SBC</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GoGrid</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>OpenNebula</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Rackspace</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>RHEV-M</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>RimuHosting</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Terremark</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>vSphere</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
HTTP client

Deltacloud

REST API

Deltacloud

Cloud provider APIs

Back End Drivers

IAAS cloud providers

- on your local machine
- in your server room
- on your network
- a public Deltacloud instance
Deltacloud Resources

- Images
- Instances
- Realms
- Keys
- Storage_volume
- Storage_snapshot
- Bucket
- Blob
- Address
- Load_balancer
- Firewall
HTTP client
- web browser
- Deltacloud ruby client
- curl
- your custom client
- any other client that can make HTTP requests

Deltacloud

Cloud provider APIs
- on your local machine
- in your server room
- on your network
- a public Deltacloud instance

IAAS cloud providers
Composer

- Imagefactory
  - Builds images from a template
- Oz
  - Automates installs
- Image Warehouse
  - Stores images locally and transfers them to providers
Imagefactory Builds

• Requires
  – name of the target cloud provider
  – template XML, a description of the image to be built

• Can build an image in two ways
Imagefactory

- Creates a raw image from scratch, using a boot image, runs the native installer
- Applies user customizations, remove and/or add software
- Uses Oz to automate the installation
Conversion

• Creates a final image that is compatible with target cloud provider

• Example
  - VMware stream-optimized format
  - qcow2 format for RHEV
Snapshot Builds

- Available with Amazon EC2
- Takes an existing JEOS in S3, makes a copy of it
- Boots it and modifies it until template is satisfied
- Saves time and bandwidth costs
Image Warehouse

- Stores images and their metadata
- Responsible for transmitting image to the cloud provider
- For Amazon EC2, bundled in S3, and registered in the correct region
- For RHEV, to the data center's storage domain
aeolus-image cli tool

[~]# aeolus-image build --target ec2,vsphere,rhevm --template template.xml

[~]# aeolus-image push --account acctname --image image_id

[~]# aeolus-image status --targetimage target_image_id

[~]# aeolus-image list --images,accounts,targets

[~]# aeolus-image import --account my-ec2 --id ami_id

[~]# aeolus-image delete --image image_id
Orchestrator

• Construct a multi-tier application that runs on multiple instances and have it self configure when it is deployed

• Example
  - Web application with MySQL as database, each on its own instance
  - Web application should start after the database is up
Deployment XML Recipe

- MySQL
  - User Param: username, password
  - Runtime Param: dbup

- Apache
  - User Param: username, password
  - Dependency to MySQL username, password, and dbup
Audrey

- Configuration server placed in a cloud provider
- Audrey agent installed in the instance
- Conductor starts up instances and sends deployment information to the configuration server
1. Instances boot and agents contact the config server

2. Config server sends MySQL Agent its parameter values
3. MySQL instance finishes loading, sets dbup fact to true and notifies config server. Config server sends Apache agent configs.

4. Apache starts up and configuration completes on the instance and agent notifies config server.

All done!
More info?

- www.aeolusproject.org
- source: https://github.com/aeolusproject
- aeolus-devel@lists.fedorahosted.org
- IRC
  Network: Freenode (irc.freenode.net)
  Channel: #aeolus