Operating and managing an Atomic container-based infrastructure

Scott Collier
Sr. Principal System Engineer

Lars Kellogg-Stedman
Senior Software Engineer

Brett Thurber
Principal Software Engineer

6/25/15
**Agenda**

**Overview**
- What is RHEL Atomic?
- Where can I get it?

**Super Privileged Containers**
- Base Images
- Utility Images

**Staying up to Date**
- Updating an Atomic Host

**Deployment Options**
- Cloud-init
- RHEV
- OpenStack

**Satellite**
- Containers

**Wrap-Up**
- Summary, questions, comments and feedback
What is Atomic?

- RHEL Atomic is a variation of Red Hat Enterprise Linux 7
- Components
  - systemd
  - Kubernetes
  - Docker
  - SELinux
- JEOS for Containers
- OSTree vs. Yum / RPM
- Everything is a Container
CONTAINER ORCHESTRATION, SCHEDULING, AND MANAGEMENT VIA KUBERNETES

- Orchestrate application services that span multiple containers across multiple Linux hosts
- Schedule containers across multiple hosts in desired topology
  - Enable manual and automated scaling up & down
- Manage container lifecycle with declarative model for health management to detect and restart on failure
Atomic Features

OSTree

OSTree: Upgrade and Rollback

atomic upgrades and rollback for the operating system

OSTree: Atomic Operating System Changes

A  or  B
Super Privileged Containers
Base and Utility Images

CONTAINER IMAGES
(RHEL 7 and RHEL Atomic Host 7)

ATOMIC CONTAINER IMAGES

RED HAT® ENTERPRISE LINUX® ATOMIC HOST

#redhat #rhsummit
Staying Up to Date
Atomic Host Upgrade

• Before
$ atomic host status

<table>
<thead>
<tr>
<th>TIMESTAMP (UTC)</th>
<th>VERSION</th>
<th>ID</th>
<th>OSNAME</th>
<th>REFSPEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-04-02 20:14:06</td>
<td>7.1.1-1</td>
<td>21bd99f9f3</td>
<td>rhel-atomic-host rhel-atomic-host:rhel-atomic-host/7/x86_64/standard</td>
<td></td>
</tr>
</tbody>
</table>

• Upgrade
$ subscription-manager register --username <username> --password <password> --auto-attach
$ atomic host upgrade && reboot

• After
$ atomic host status

<table>
<thead>
<tr>
<th>TIMESTAMP (UTC)</th>
<th>VERSION</th>
<th>ID</th>
<th>OSNAME</th>
<th>REFSPEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015-05-07 19:00:48</td>
<td>7.1.2</td>
<td>203dd666d3</td>
<td>rhel-atomic-host rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard</td>
<td></td>
</tr>
<tr>
<td>2015-04-02 20:14:06</td>
<td>7.1.1-1</td>
<td>21bd99f9f3</td>
<td>rhel-atomic-host rhel-atomic-host-ostree:rhel-atomic-host/7/x86_64/standard</td>
<td></td>
</tr>
</tbody>
</table>
Infrastructure Deployment Options
Where can I get this cool software?
Obtaining Atomic

Red Hat Customer Portal

- Cloud Image
- RHEV Image
- Hyper-V Image
- vSphere Image
- Installer ISO
What can I deploy on?

- GCE
- AWS
- KVM
- VMware

- etc...

- Bare Metal
- PXE
- RHEV
- OpenStack
Deployment Options - RHEV
Deployment Options - OpenStack

Manual deployment for a single server (or small number of servers)

OR

Automated deployment using OpenStack Heat
Configuration with cloud-init

- Reads “metadata” (hostname, ssh public keys, etc) and “user data” (arbitrary shell scripts and other configuration data)

- Information may be sourced from network (the “metadata service”), from a “configuration drive”, or may be baked into your disk image.

- Performs a variety of initial system configuration tasks

```bash
#!/bin/sh
sed -i '^
^SELINUX=' s/=.*/=enforcing/
/etc/selinux/config
setenforce 1
```

```bash
#cloud-config
packages:
- git
- httpd
```
Manual Deployment

$ nova boot \
  --image rhel-atomic-20150615 \ 
  --key-name lars --flavor m1.small \ 
  --user-data my-script.sh \ 
  --nic net-id=... atomic
Using OpenStack Heat to deploy a cluster

- Templates
- Parameters
- Heat
  - Nova
  - Cinder
  - Neutron
  - Ceilometer
Heat templates

Major sections:

- parameters
  - input into your templates
- resources
  - describes the resources Heat will create
- outputs
  - Information derived from the deployed stack

```yaml
heat_template_version: 2014-10-16
description: >
  This is an example.

parameters:

resources:

outputs:
```
heat_template_version: 2014-10-16

resources:

  node:
    type: OS::Nova::Server
    properties:
      image: rhel-atomic-20150615
      flavor: m1.small
      key_name: lars
      networks:
        - port: {get_resource: node_eth0}

node_eth0:
  type: OS::Neutron::Port
  properties:
    network: net0
    security_groups:
      - default
    fixed_ips:
      - subnet: net0-subnet0

node_floating:
  type: OS::Neutron::FloatingIP
  properties:
    floating_network: public
    port_id: {get_resource: node_eth0}
Heat-Kubernetes Templates

https://github.com/projectatomic/heat-kubernetes/
Using the Templates

Clone and deploy:

```bash
git clone https://github.com/projectatomic/heat-kubernetes
cd heat-kubernetes
heat stack-create -f kubecluster.yaml -e local.yaml my-kube-cluster
```

Deploying directly from GitHub:

```bash
heat stack-create -f https://raw.githubusercontent.com/projectatomic/heat-kubernetes/master/kubecluster.yaml -e local.yaml my-kube-cluster
```
Using the Templates

Providing required (and optional) parameters:

```yaml
parameters:
  ssh_key_name: lars
  server_image: rhel-atomic-20150615
  dns_nameserver: 192.168.122.1
  docker_volume_size: 5
```
Demo
Satellite 6 with Containers

- Deploy RHEL Atomic Host
- Atomic Host Preparation
- Atomic Host Compute Resource
- Content Management
- Container Deployment
Deploy RHEL Atomic Host From Satellite

• Although not explicitly supported at this time it is possible and planned in future Satellite releases

• Reference blog for deployment: https://access.redhat.com/blogs/1169563/posts/1318283
Preparing the RHEL Atomic Host

- Install the Satellite certificate to the RHEL Atomic host
- Copy the .crt and .pem files to the appropriate locations
- Execute `update-ca-trust` to trust the installed cert
- Restart docker for changes to take effect
- Register the atomic host with the Satellite server via `subscription-manager`
RHEL Atomic host as a compute resource in Satellite

- Add the RHEL Atomic host as a compute resource in Satellite

New compute resource

<table>
<thead>
<tr>
<th>Compute Resource</th>
<th>Locations</th>
<th>Organizations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name *</td>
<td>Red Hat Atomic</td>
<td></td>
</tr>
<tr>
<td>Provider *</td>
<td>Docker</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>URL *</td>
<td><a href="http://rmi-atomic.rearch.bos.redhat.com:2375">http://rmi-atomic.rearch.bos.redhat.com:2375</a></td>
<td>e.g. <a href="https://docker.example.com:4243">https://docker.example.com:4243</a> or <a href="">unix:///var/run/docker.sock</a></td>
</tr>
</tbody>
</table>
Content Management

- Multiple options exist within Satellite for Red Hat docker image content:
  - **Custom product**
    - Used to host docker content locally
    - Created as a new product with an associated repository
    - Added and promoted to a content view, associated to a lifecycle environment
  - **External docker registries**
    - Red Hat
    - custom
Container Deployment

- There are several options to choose from when deploying containers:
  - *(From Satellite)* Content View
    - Utilizes locally stored content
  - *(From Satellite)* External Registry
    - Utilizes externally hosted content
  - *(From RHEL Atomic Host)* `docker pull` and `docker run`
    - Pulling and running images from Satellite
Demo
Summary
Summary

- Deployment Options
- Updates
- Providers
- Utility Images
- Cloud-init
- OpenStack
- Satellite Integration