What's New In 5.2

What’s in the Chimichanga

- Lots of internal changes
- “Attributes” in the graph
  - Also can have “edge conditionals”
- Isolated MySQL (finally!)
  - “yum update” won’t destroy the Rocks database
- Solaris-based backend nodes

Attributes

- Assign values to variables in the graph
- An evolution of the <var> tags and the app_globals table
- Previous syntax:

```
ServerName <var name="Kickstart_PublicHostname"/>
```

- New syntax:

```
ServerName &Kickstart_PublicHostname;
```

Attributes

- Attributes can be set at 4 levels:
  - Globally
    - ‘rocks set attr’
  - By appliance type
    - ‘rocks set appliance attr’
  - By OS (linux or sunos)
    - ‘rocks set os attr’
  - By host
    - ‘rocks set host attr’

Attributes

- Example, set the public IP address of a remote frontend that is used during a ‘central’ installation:

```
# rocks set host attr vi-1.rocksclusters.org \
  Kickstart_PublicAddress 137.110.119.118
```

Attributes

```
# rocks list host attr tile-0-0
HOST      ATTR                                  VALUE                                    SOURCE
tile-0-0: Info_CertificateCountry               US                                       G
tile-0-0: Info_CertificateLocality              San Diego                                G
tile-0-0: Info_CertificateOrganization          CalIT2                                   G
tile-0-0: Kickstart_DistroDir                   /export/rocks                            G
tile-0-0: Kickstart_PrivateAddress              10.1.1.1                                 G
tile-0-0: Kickstart_PrivateBroadcast            10.1.255.255                             G
tile-0-0: Kickstart_PrivateDNSDomain            local                                    G
tile-0-0: Kickstart_PrivateDNSServers           10.1.1.1                                 G
tile-0-0: Kickstart_PrivateGateway              10.1.1.1                                 G
tile-0-0: Kickstart_PublicDNSServers            132.239.0.252                            G
tile-0-0: Kickstart_PublicGateway               137.110.119.1                            G
tile-0-0: Kickstart_PublicHostname              vizagra.rocksclusters.org                G
tile-0-0: Kickstart_PublicKickstartHost         central.rocksclusters.org                G
tile-0-0: Kickstart_PublicNTPHost               pool.ntp.org                             G
tile-0-0: Kickstart_PublicNetmask               255.255.255.0                            G
tile-0-0: Kickstart_PublicNetmaskCIDR           24                                       G
tile-0-0: Kickstart_PublicNetwork               137.110.119.0                            G
tile-0-0: Kickstart_Timezone                    America/Los_Angeles                      G
tile-0-0: Server_Partitioning                   force-default-root-disk-only             G
tile-0-0: arch                                  x86_64                                   H
tile-0-0: hostname                              tile-0-0                                 I
tile-0-0: rack                                  0                                        I
tile-0-0: rank                                  0                                        I
tile-0-0: rocks_version                         5.2                                      G
tile-0-0: HideBezels                            false                                    G
tile-0-0: HttpConf                              /etc/httpd/conf                          O
tile-0-0: HttpConfigDirExt                      /etc/httpd/conf.d                        O
tile-0-0: HttpRoot                              /var/www/html                            O
```

Edge Conditionals

- Use attributes to conditionally traverse edges of the configuration graph

```
<edge from="client" cond="rsh">
      <to>rsh</to>
</edge>
```

- If ‘rsh’ evaluates to ‘true’, then the edge from ‘client’ to ‘rsh’ will be traversed
  - Default value is ‘false’

Edge Conditionals

- To set a conditional attribute:

```
# rocks set attr rsh true
```

- Edge conditionals are attributes
  - Can also be set at 4 levels:
    - Globally
    - By appliance type
    - By OS (linux or sunos)
    - By host

Route Commands

- Did you know there was a ‘route’ table in the database?
- We finally added rocks commands to configure routes
- Example: add a global route for multicast traffic:

```
# /opt/rocks/bin/rocks add route 224.0.0.0 eth0 \
     netmask=255.255.255.0
```

Route Commands

- Routes can be added:
  - Globally
  - By appliance type
  - By OS (linux or sunos)
  - By host

```
# /opt/rocks/bin/rocks add route 224.0.0.0 eth0 \
    netmask=255.255.255.0

# /opt/rocks/bin/rocks add host route bayou 0.0.0.0 \
        137.110.119.1 netmask=0.0.0.0
```

Alias Commands

- Did you know there was an ‘aliases’ table in the database?
- We finally added rocks commands to configure host aliases

Alias Commands

- Example:

```
# rocks add host alias vm-container-0-0 v-0-0

# cat /etc/hosts
127.0.0.1    localhost.localdomain      localhost
10.1.1.1     bayou.local bayou # originally frontend-0-0
10.1.255.254 vm-container-0-0.local vm-container-0-0 v-0-0
10.1.255.253 vm-container-0-1.local vm-container-0-1
10.1.255.252 frontend-0-0-0.local frontend-0-0-0
```

Foundation MySQL

- MySQL for Rocks is now isolated
  - Installed under /opt/rocks

- When Tim Carlson does a “yum update”, he now won’t hose the Rocks database

IPMI Support

- Rocks commands to configure IPMI subnets
  - Create IPMI network (like “private” and “public”)

  - /etc/sysconfig/network-scripts/ipmi-X
    - Where ‘X’ is the channel

Multi-Version and Multi-Architecture Support

- Vmlinuz and initrd.img are versioned with Rocks release and architecture

```
# rocks list bootaction output-col='action,kernel,ramdisk'
ACTION               KERNEL                 RAMDISK
install:             vmlinuz-5.2-x86_64     initrd.img-5.2-x86_64
```

- Enables supporting 64-bit and 32-bit physical and virtual machines
- Going forward, we should be able to support multiple version of Rocks on the backend nodes
  - The trick will be how the distribution is built

Retooled Boot Action

- In Rocks v5.1 had two different ways to instruct physical and virtual hosts what to do on their next boot:
  - Physical

    ```
    # rocks set host boot pxeboot comptue-0-0 action=“install”
    ```

  - Virtual

    ```
    # rocks set host vm bootprofile profile=install \
      kernel="file:///boot/kickstart/xen/vmlinuz"
    # rocks start host vm hosted-vm-0-0 install=y
    ```

- Now:

  ```
  # rocks set host boot compute-0-0 action=“install”
  # rocks set host boot hosted-vm-0-0 action=“install”
  ```

Tweaked Xen Roll Internals

- All Xen commands are issued with “libvirt”
  - Makes the Xen Roll more VM agnostic
  - Virtualization management calls are much faster
    - Previous release, all VM management commands were issued via an ssh tunnel.
  - Should be easy to support Xen, KVM, QEMU, etc.
  - Significant because Red Hat has announced they are moving away from Xen

- “Lights out” VM frontend install
  - Fully-automated VM frontend install
    - Don’t have to enter data at the user input screens
  - Full-automated physical frontends should work too
    - Just haven’t tried it yet

Solaris Release

- Rocks v5.2 supports Solaris-based backend nodes
- Solaris code has been merged with the head of the Rocks tree