Isn’t it Ironic?
Managing a *bare metal* cloud

Devananda van der Veen
twitter: @devananda

devananda.github.io/talks/isnt-it-ironic.html
Who am I

- Master Engineer at HP
- OpenStack Ironic PTL
- OpenStack Technical Committee
What we’re going to talk about

- Virtualization & OpenStack
- Ironic's architecture
- Configuration choices you need to make
- Operations
- Limitations
- Walk through a deploy
"OpenStack is not a virtualization layer. It is an abstraction layer."

- Daniel Sabbah, CTO @ IBM
Google trends:
Virtualization & Cloud Computing
Google trends:
DevOps & OpenStack
What do developers really want?

- Separate application delivery from hardware procurement
- Self-service API for Compute [Network, Storage] resources
- More control + More flexibility
So what is this Ironic thing, anyway?

- Python services that abstract hardware management
- Consistent API across server vendors
- Integrate with OpenStack
  - or run Ironic by itself!
Key Components

- **ironic-api**: RESTful API service
- **ironic-conductor**: interacts directly with hardware; asynchronous handling of both requested and periodic actions.
- **ironic-python-agent**: utility service, temporarily booted on machines to provide remote access to hardware for provisioning and management.
- **Nova driver**: interface for Nova; enables OpenStack to provide common abstraction for virtual and physical machines.
- **discoverd ramdisk**: optional tool for hardware inventory management.
- **bifrost**: ansible modules for getting started with Ironic outside of OpenStack.
Open Technologies

- **IPMI**: intelligent platform management interface, for remote control of machine power state, boot device, serial console, etc.
- **DHCP**: dynamic host configuration protocol, used to locate the NBP on the network, and provide the host OS with IP address during init
- **TFTP**: trivial file transfer protocol, copies the NBP over the network
- **PXE**: pre-boot execution environment, allows host to boot from network
- **iPXE**: recent enhancements make PXE more flexible, supported on most hardware
- **iSCSI**: used to remotely attach to HDD and copy the machine image
What about Vendor-specific enhancements?

Yes!

SeaMicro, Dell, Fujitsu, HP, IBM, Intel, OpenCompute, Cisco, ...
And so you have options...

- **IPMI**: vendor-specific power management; varies by vendor
- **DHCP**: static IP injection is possible, but not suitable for larger or dynamic environments
- **PXE**: boot over virtual media channel; support varies by vendor
- **iSCSI**: user image can be fetched directly by "agent" drivers
... and options ...

- **Homogeneous hardware?**
  Easy!

- **Heterogeneous hardware?**
  Use nova-scheduler to match flavor $\leftarrow\rightarrow$ node.properties

- **Single tenant / small deployment?**
  Flat network. Maybe use Ironic stand-alone

- **Service provider for multiple tenants?**
  Use Keystone for auth, Nova for quota management, Neutron for net isolation (*)
  Basically, use OpenStack

- **Untrusted tenants?**
  Network isolation is possible via Neutron
  Secure-erase disks, flash firmware between each use
  (Some assembly required)
New in Kilo:

- Instances may **boot from local disk** with all drivers
- Local **configdrives** remove dependence on meta-data service
- **Secure-erase** disk drives between each use
- API **version headers** improve compatibility during upgrades
- Nodes may be addressed by **logical names** in addition to UUIDs
- Drivers may store **internal attributes** and can register their own **periodic tasks**
Operations

- Configuration
- Building Images
- Limitations
Nova Configuration

[default]

# Driver to use for controlling virtualization. Options
compute_driver=nova.virt.ironic.IronicDriver

# Firewall driver (defaults to hypervisor specific iptables driver)
firewall_driver=nova.virt.firewall.NoopFirewallDriver

# The scheduler host manager class to use (string value)
scheduler_host_manager=nova.scheduler.ironic_host_manager.IronicHostManager

# Virtual ram to physical ram allocation ratio which affects
# all ram filters. This configuration specifies a global ratio
ram_allocation_ratio=1.0

# Amount of disk in MB to reserve for the host (integer value)
reserved_host_memory_mb=0

# Full class name for the Manager for compute (string value)
compute_manager=ironic.nova.compute.manager.ClusteredComputeManager
[ironic]

# Ironic keystone admin name
admin_username=ironic

# Ironic keystone admin password.
admin_password=IRONIC_PASSWORD

# keystone API endpoint
admin_url=http://IDENTITY_IP:35357/v2.0

# Ironic keystone tenant name.
admin_tenant_name=service

# URL for Ironic API endpoint.
api_endpoint=http://IRONIC_NODE:6385/v1
Building Your Machine Images with diskimage-builder

disk-image-create -a amd64 -o my-image -t qcow2 \
  vm ubuntu serial-console cloud-init-datasources

glance image-create --name my-image --is-public True \ 
  --disk-format qcow2 --container-format bare < my-image.qcow2
Managing Nova Flavors
Create the flavor

```
nova flavor-create my-baremetal-flavor auto $RAM_MB $DISK_GB $CPU
```

Setting additional hints

```
ironic node-update add properties/capabilities='boot_mode:uefi'
nova flavor-key my-baremetal-flavor set capabilities:boot_mode='uefi'
```
Step 1. Operator enrols hardware

$ ironic node-create ...

Ironic API

Ironic Conductor
- hardware drivers

servers

---

Step 2. User creates instance

$ nova boot ...

Nova Compute
- Ironic virt driver

Nova Scheduler

Nova API
Limitations

- **Firmware and RAID**
  Plugin framework exists in ironic-python-agent, but... Today, you must BYO plugin

- **NICs <-> Networks**
  Nova only supports one-to-one mapping today

- **Provisioning Network <-> Tenant Network Separation**
  Upstream only supports flat network today. Out-of-tree options exist; being upstrommed now

- **Per-tenant Network Isolation**
  No official support today; several solutions proposed. Work with Neutron is happening now
Examples or Demo?
Enroll Hardware

```bash
$ ironic node-create -d agent_ipmitool \
  -i ipmi_username=admin -i ipmi_password=fake -i ipmi_address=10.1.2.3 \
  -p cpus=4 -p memory_mb=8192 -p local_gb=500 \
  -e note='spare server' -n mytest
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassis_uuid</td>
<td>None</td>
</tr>
<tr>
<td>driver</td>
<td>agent_ipmitool</td>
</tr>
<tr>
<td>driver_info</td>
<td>{'ipmi_address': '10.1.2.3', 'ipmi_username': 'admin', 'ipmi_password': '******'}</td>
</tr>
<tr>
<td>extra</td>
<td>{}</td>
</tr>
<tr>
<td>properties</td>
<td>{'memory_mb': '8192', 'local_gb': '500', 'cpus': '4'}</td>
</tr>
<tr>
<td>uuid</td>
<td>7a1ce8d0-9679-4d87-8f54-b11f6e8adb8f</td>
</tr>
<tr>
<td>name</td>
<td>mytest</td>
</tr>
</tbody>
</table>

```bash
$ ironic port-create -n 7a1ce8d0-9679-4d87-8f54-b11f6e8adb8f -a 00:11:22:00:11:22
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_uuid</td>
<td>7a1ce8d0-9679-4d87-8f54-b11f6e8adb8f</td>
</tr>
<tr>
<td>extra</td>
<td>{}</td>
</tr>
<tr>
<td>uuid</td>
<td>024e52b2-6ae4-483b-a039-d6afae7f6a22</td>
</tr>
<tr>
<td>address</td>
<td>00:11:22:00:11:22</td>
</tr>
</tbody>
</table>
Validate provided info

```
$ ironic node-validate 7a1ce8d0-9679-4d87-8f54-b11f6e8adb8f
+---------------------------------+-------------------+----------------------------------+
| Interface | Result | Reason                                                                                   |
+---------------------------------+-------------------+------------------------------------------------------------------------------------------|
| console | False | Missing 'ipmi_terminal_port' parameter in node's driver_info.                             |
| deploy   | False | Node 7a1ce8d0-9679-4d87-8f54-b11f6e8adb8f failed to validate deploy image info. Some parameters were missing. Missing are:
|         |       | ['driver_info.deploy_kernel', 'driver_info.deploy_ramdisk', 'instance_info.image_source'] |
| inspect | None  | not supported                                                                             |
| management | True |                                                                                           |
| power   | True  |                                                                                           |
```
Oops
(I forgot a few options)
Add or change options

```
$ ironic node-update mytest add \ 
    instance_info/image_source=http://192.168.1.1/myimage.qcow2 \ 
    instance_info/image_checksum=e1d99d6d0ef2144a8d672b0420c547b5

$ ironic node-update mytest add \ 
    driver_info/deploy_ramdisk=http://192.168.1.1/deploy.initrd \ 
    driver_info/deploy_kernel=http://192.168.1.1/deploy.vmlinux

$ ironic node-update mytest replace extra/note='database' name=db01.example

+-------------------+---------------------+
| Property          | Value               |
+-------------------+---------------------+
| extra             | {u'note': u'database'} |
| name              | db01.example        |
```
Validate info (again)

$ ironic node-validate db01.example

<table>
<thead>
<tr>
<th>Interface</th>
<th>Result</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>False</td>
<td>Missing 'ipmi_terminal_port' parameter in node's driver_info.</td>
</tr>
<tr>
<td>deploy</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>inspect</td>
<td>None</td>
<td>not supported</td>
</tr>
<tr>
<td>management</td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>power</td>
<td>True</td>
<td></td>
</tr>
</tbody>
</table>
$ ironic node-show db01.example

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>target_power_state</td>
<td>None</td>
</tr>
<tr>
<td>last_error</td>
<td></td>
</tr>
<tr>
<td>maintenance_reason</td>
<td></td>
</tr>
<tr>
<td>provision_state</td>
<td>available</td>
</tr>
<tr>
<td>console_enabled</td>
<td>False</td>
</tr>
<tr>
<td>target_provision_state</td>
<td>None</td>
</tr>
<tr>
<td>maintenance</td>
<td>False</td>
</tr>
<tr>
<td>power_state</td>
<td>power off</td>
</tr>
<tr>
<td>driver</td>
<td>agent_ipmitool</td>
</tr>
<tr>
<td>reservation</td>
<td>None</td>
</tr>
<tr>
<td>instance_uuid</td>
<td>None</td>
</tr>
<tr>
<td>driver_internal_info</td>
<td>{}</td>
</tr>
<tr>
<td>chassis_uuid</td>
<td></td>
</tr>
</tbody>
</table>
### Maintenance Mode

```bash
$ ironic node-set-maintenance --reason 'replacing disks' db01.example true
$ ironic node-show db01.example
```

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>target_power_state</td>
<td>None</td>
</tr>
<tr>
<td>last_error</td>
<td>None</td>
</tr>
<tr>
<td>maintenance_reason</td>
<td>replacing disks</td>
</tr>
<tr>
<td>provision_state</td>
<td>available</td>
</tr>
<tr>
<td>console_enabled</td>
<td>False</td>
</tr>
<tr>
<td>target_provision_state</td>
<td>None</td>
</tr>
<tr>
<td>maintenance</td>
<td>True</td>
</tr>
<tr>
<td>power_state</td>
<td>power off</td>
</tr>
<tr>
<td>instance_uuid</td>
<td>None</td>
</tr>
<tr>
<td>driver_internal_info</td>
<td>{}</td>
</tr>
</tbody>
</table>
Power Status Loop

```
$ ironic node-show my.broken.node
  +-----------------+-----------------
  | Property        | Value
  +-----------------+-----------------
  | last_error      | During sync_power_state, max retries exceeded for node
  |                 | 9729f0b2-b270-4d06-aa87-40f2b2cad6ee, node state None does not match expected state 'off'. Updating DB state to 'None' Switching node to maintenance mode.

$ cat /var/log/upstart/ironic-conductor.log
2015-03-24 04:29:19.349 26317 WARNING ironic.conductor.manager [-]
During sync_power_state, could not get power state for node 9729f0b2-b270-4d06-aa87-40f2b2cad6ee, node state None does not match expected state 'off'. Updating DB state to 'None' Switching node to maintenance mode.
```

Deployment (via Ironic)

$ ironic node-set-provision-state db01.example active

    The provisioning operation can't be performed on node
    7a1ce8d0-9679-4d87-8f54-b11f6e8adb8f because it's in maintenance mode.

$ ironic node-set-maintenance db01.example false
$ ironic node-set-provision-state db01.example active
$ # ... time goes on ...
## Deployment (via Ironic)

```
$ ironic node-show db01.example

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>target_power_state</td>
<td>None</td>
</tr>
<tr>
<td>last_error</td>
<td></td>
</tr>
<tr>
<td>maintenance_reason</td>
<td>None</td>
</tr>
<tr>
<td>provision_state</td>
<td>active</td>
</tr>
<tr>
<td>console_enabled</td>
<td>False</td>
</tr>
<tr>
<td>target_provision_state</td>
<td>None</td>
</tr>
<tr>
<td>maintenance</td>
<td>False</td>
</tr>
<tr>
<td>power_state</td>
<td>power on</td>
</tr>
<tr>
<td>instance_uuid</td>
<td>None</td>
</tr>
<tr>
<td>driver_internal_info</td>
<td>{}</td>
</tr>
</tbody>
</table>
```
Deployment (via Nova)

```
$ nova boot --flavor baremetal -image myimage -key-name my_ssh_key ...

$ tail -f /var/log/upsart/nova-compute.log
...
2014-05-01 03:47:05.878 AUDIT nova.compute.resource_tracker [-] Free ram (MB): 8192
2014-05-01 03:47:05.878 AUDIT nova.compute.resource_tracker [-] Free disk (GB): 500
2014-05-01 03:47:05.878 AUDIT nova.compute.resource_tracker [-] Free VCPUS: 4
...
2014-05-01 03:47:05.878 AUDIT nova.compute.resource_tracker [-] Free ram (MB): 0
2014-05-01 03:47:05.878 AUDIT nova.compute.resource_tracker [-] Free disk (GB): 0
2014-05-01 03:47:05.878 AUDIT nova.compute.resource_tracker [-] Free VCPUS: 0
```
Two methods for image deployment

Direct from source || Cache on conductor

- agent_ipmitool
- agent_pyghmi
- agent_ilo
- pxe_ipmitool
- pxe_ipminative
- pxe_seamicro
- pxe_iboot
- pxe_ilo
- pxe_snmp
- pxe_drac
- pxe_irmc
- pxe_amt
- iscsi_ilo
PXE Deploy Process (cont)
Agent Deploy Process

- Nova
  - set instance info
  - set provision state

- API
  - do_node_deploy()
  - update pxe, ftp config

- Conductor
  - update DHCPBOOT
  - power on

- Neutron
  - DHCP request
  - next-server = Conductor
  - attempts tftpboot
  - send agent kernel, ramdisk, & config
  - runs agent ramdisk

- Node
  - lookup()
  - pass UUID
  - heartbeat(UUID)
  - heartbeat
Thanks!

@devananda

devananda.github.io/talks/isnt-it-ironic.html
docs.openstack.org/developer/ironic/deploy/install-guide.html

Give us feedback!
Ops track // Wednesday 9:50am room 216 // http://sched.co/3Rca