Bare-metal as a service

OpenStack Ironic service deployment

Cedric Morandin / Virginie Longo
Cloud engineers
05/08/2018
Our environment
Amadeus in few words

- **630+ million total bookings processed in 2017** using the Amadeus distribution platform
- **1.6 billion** passengers boarded in 2017 with Amadeus and Navitaire solutions
- **1 of the world’s top 15 software companies**
  Forbes 2017 global rankings
Why OpenStack
For the R&D environment

Developers want a platform allowing them to quickly prototype
And a safe place where to test new technologies

They want to easily spawn development and test platforms
And automate infrastructure deployment
R&D cloud platform anatomy

- OpenStack
- OSP8
- Software Defined Network
- Nuage
- Software Defined Storage
- Ceph
- Hypervisor
- KVM
Under the hood
The platform

LOGICAL

VM  VM  VM  VM  VM  VM

VSG

CLIENTS

DATA

PHYSICAL

Ceph storage

control plane

Amadeus network

API

VM

VM

VM

VM

VM

VM
What do we want?

LOGICAL

VM VM VM VM VM VM

bare-metal

VSG

Amadeus network

CLIENTS

DATA

PHYSICAL

Ceph storage

control plane

API

VM VM VM VM VM VM
Introducing Ironic

“Ironic aims to provision bare metal machines instead of virtual machines, forked from the Nova bare-metal driver. By default, it will use PXE and IPMI in concert to provision and turn on/off machines.”
Why using Ironic
and what are the requirements?

- **Performance testing** of applications
- Hosting applications with **built-in redundancy**
- **Multi-tenancy** for bare-metal
- Standard **Nova API** for manipulating bare-metal servers
- Images stored in **Glance**
How does it work?

https://docs.openstack.org/ironic/pike/user/ - section 1.2
Implementation and usage
Let’s deploy Ironic!

Liberty version

#1 Configure Nuage SDN
Let’s deploy Ironic!

Liberty version

#1 Configure Nuage SDN

#2 Deploy a VM with nova compute and all Ironic services
Let’s deploy Ironic!

Liberty version

#1 Configure Nuage SDN

#2 Deploy a VM with nova compute and all Ironic services

#3 Reconfigure OpenStack controllers
Let’s deploy Ironic!
Liberty version

#1 Configure Nuage SDN

#2 Deploy a VM with nova compute and all Ironic services

#3 Reconfigure OpenStack controllers

#4 Create Keystone service catalog entries for OpenStack service
Let’s configure Ironic!

#5 Register and configure the bare-metal nodes
Let’s configure Ironic!

#5 Register and configure the bare-metal nodes

$ openstack baremetal create nceospp01bms663.yaml
$ openstack baremetal node set --property capabilities=boot_option:local\
--driver-info deploy_kernel=de99de5d-341d-4c8a-bc72-2cd04416f77f\
--driver-info deploy_ramdisk=86a2aa1f-3b04-4f3d-9330-dcac08a283e8 nceospp01bms663

$ openstack baremetal port set dd7428df-3f8b-4867-a633-d590ebade831 --extra\
gateway_name=10.255.110.248 --extra gateway_port=1/1/5 --extra gateway_vlan=0
Let’s configure Ironic!

#5 Register and configure the bare-metal nodes

$ openstack baremetal create nceospp01bms663.yaml
$ openstack baremetal node set --property_capabilities=boot_option:local
--driver-info deploy_kernel=de99de5d-341d-4c8a-bc72-2cd04416f77f
--driver-info deploy_ramdisk=86a2aa1f-3b04-4f3d-9330-dcac08a283e8 nceospp01bms663

$ openstack baremetal port set dd7428df-3f8b-4867-a633-d590ebade831 --extra
gateway_name=10.255.110.248 --extra gateway_port=1/1/5 --extra gateway_vlan=0

$ ironic node-list --fields name power_state provisioning_state instance_uuid

<table>
<thead>
<tr>
<th>Name</th>
<th>Power State</th>
<th>Provisioning State</th>
<th>Instance UUID</th>
</tr>
</thead>
<tbody>
<tr>
<td>nceospp01bms651</td>
<td>power on</td>
<td>active</td>
<td>79148cc7-f375-4b98-a5f0-fe96c587b406</td>
</tr>
<tr>
<td>nceospp01bms663</td>
<td>power off</td>
<td>available</td>
<td>None</td>
</tr>
<tr>
<td>nceospp01bms664</td>
<td>power on</td>
<td>deploying</td>
<td>a4dc68ab-9bbe-4ff7-b5a5-d18df81f7400</td>
</tr>
</tbody>
</table>

#redhat #rhsummit
Let’s configure Ironic!

#6 Create the appropriate flavor
Let’s configure Ironic!

#6 Create the appropriate flavor

(openstack) flavor show bm1.c6420

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>200</td>
</tr>
<tr>
<td>id</td>
<td>7e961875-3cae-4e4d-ac3a-ca8e369f010b</td>
</tr>
<tr>
<td>name</td>
<td>bm1.c6420</td>
</tr>
<tr>
<td>os-flavor-access:is_public</td>
<td>True</td>
</tr>
<tr>
<td>properties</td>
<td>aggregate_instance_extra_specs:baremetal='true', capabilities:boot_option='local'</td>
</tr>
<tr>
<td>ram</td>
<td>60000</td>
</tr>
<tr>
<td>rxtx_factor</td>
<td>1.0</td>
</tr>
<tr>
<td>vcpus</td>
<td>10</td>
</tr>
</tbody>
</table>
Pixie boots rockin’
The first bare-metal server

$ nova boot bms-centos-raw --image centos-bms-qlogic\n--flavor bm1.c6420 --config-drive true\n--nic net-id=b08e83b8-f9b0-4922-9b63-149dc7fdc79c\n--key-name mykey
The first bare-metal server

[vlongo@bastion01 ~]$ ssh centos@172.16.156.12
[centos@jump ~]$ ssh centos@192.168.1.29 -i mykey.pem
[centos@bms-centos-raw ~]$
It works!

Multi-tenancy achieved thanks to VSGs

Easily usable by people as because integrated with Nova

Images are stored in Glance and managed as normal images
But it could be even better!

<table>
<thead>
<tr>
<th>Connectivity</th>
<th>No NAT operation possible at this time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quotas</td>
<td>No quota to limit the number of bare-metal servers per tenant</td>
</tr>
<tr>
<td>Metadata</td>
<td>Available only through config drive</td>
</tr>
<tr>
<td>Images</td>
<td>Because of the config drive being used for metadata specific images have to be built for Ironic</td>
</tr>
</tbody>
</table>
The journey continues...
What’s next?

Deployment of OSP12 (Pike) solving most of the problems

Onboarding of new customers and adding more nodes

Design a solution allowing NAT connectivity
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHat