Ironic 研究报告

谢涛涛10107965

# 简介

* 1. 功能概述

Ironic 是 openstack 的一个项目，提供物理硬件，而不是虚拟机，即常说的裸机功能。

直接操作物理硬件，不借助 hypervisor。相对于虚拟机，缺少了 hostOS。



图2.1 传统虚拟技术和裸机管理的不同

如此实现使得管理物理机如同管理虚拟机般方便。虚拟机的管理，比如部署、删除、开关机等操作，均是通过 hypervisor 的各个 API 来实现，如 libvirt driver 中 create 接口用于部署，在 ironic 中，也是借助各种各样的 driver 来实现，比如 PXE、ipmitool 等，其中 PXE 用于部署，ipmitool 用于电管理（上下电），相互配合来实现。

ironic 将会替代nova 代码 virt 中的 baremetal 驱动。

* 1. 术语解释

|  |  |
| --- | --- |
| 名称 | 解释 |
| compute host | 运行 nova-compute 服务的host |
| baremetal host | 运行 ironic 相关服务的host；注：一般情况下 compute host 和 baremetal host 合一 |
| baremetal node | 被 baremetal host 控制的服务器，可为裸机也可为已安装操作系统的服务器，此处的控制是借助于 driver，比如 IPMI 或者 SSH。需要手动注册硬件信息，比如 MAC/CPU/MEM/DISK/IPMI信息 |
| Deploy image | 用于 PXE 安装启动的 kernel 和 ramdisk 的映像文件 |
| PXE | Preboot Execution Environment |
| DHCP | Dynamic Host Configuration Protocol |
| NBP | Network Bootstrap Program |
| TFTP | Trivial File Transfer Protocol |
| IPMI | Intelligent Platform Management Interface |
|  |  |

* 1. 社区进展

从 G 版本开始替代 nova.virt 的 baremetal driver，K 版本会毕业，推荐从 J release 开始使用。

# 背景

* 1. **应用场景的需求**

存在如下的场景：

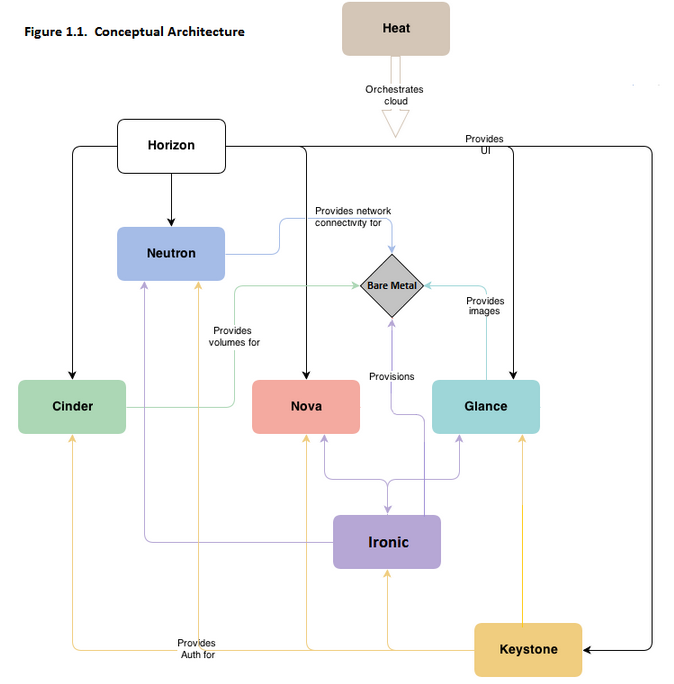
* 高性能计算集群
* 计算任务需要直接访问物理硬件，而不是虚拟化出来的
* 数据库托管，一些数据库在 hypervisor 上运行性能较差
* 单租户，专用硬件，安全，独立和其他监管需要
* 云设施的快速部署
  1. 云管理的需求

物理机和虚拟机管理有很多地方非常相似，比如物理机和虚拟机都需要开机关机，安装部署，添加和删除，为了避免重复造轮子。

# 架构

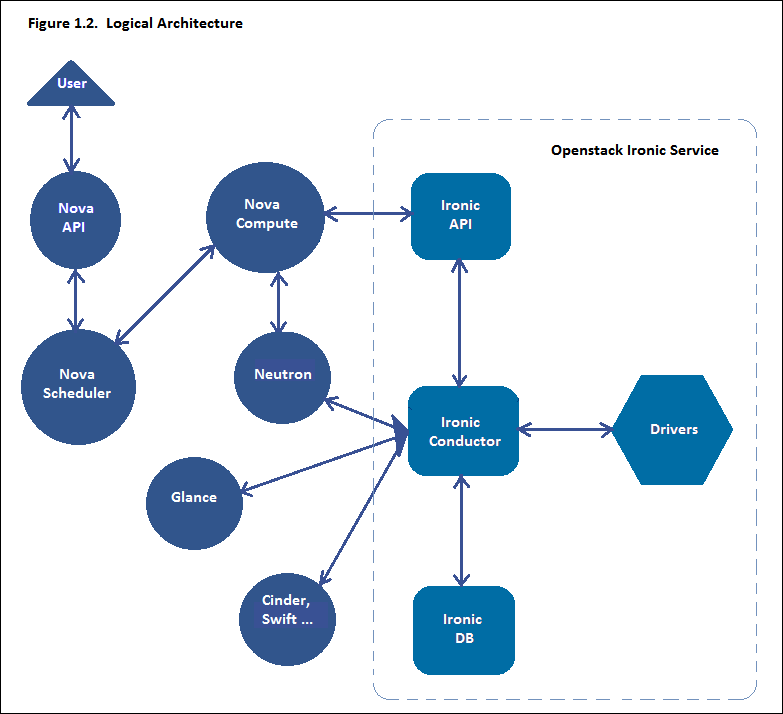
* 1. 概念架构

Note：swift 已经可以用于 ironic，此图没有标注。



* 1. 逻辑架构

下图展示了 ironic 内部服务（api/conductor/db）之间的关系，ironic 组件与其他组件（nova/neutron/glance/cinder, etc）的关系，以及 node deploy 的逻辑流程。



1. ironic 的服务包括如下几部分：

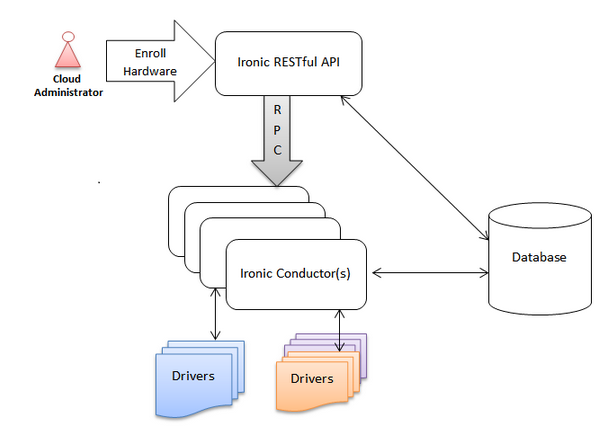
* ironic-api 服务，对外接口
* ironic-conductor 服务，真正的干活者，接收 ironic-api 的发来的“订单”，进行相应的工作。
* 消息队列，上述两者之间通信用
* 数据库 ironic，存储相关信息
* 多种 drivers

1. 从逻辑架构图中看到，部署一个 baremetal instance的流程，从 nova-api 发起 nova boot 的请求，经过 nova-scheduler 到 nova-compute，后者将请求整理发送给 ironic-api，ironic-api 再将消息下发给 ironic-conductor，ironic-conductor 中有很多的 driver，这些 driver 涵盖异构的硬件和数据库（？）等等，接着这些 driver 实现部署功能，提供物理机给用户。
2. 和部署一个虚拟机类似，ironic 也需要 openstack 其他组件的配合，从 glance 获取镜像，从 neutron 获取网络，从 cinder 获取云硬盘，等等。
3. 几种关键技术

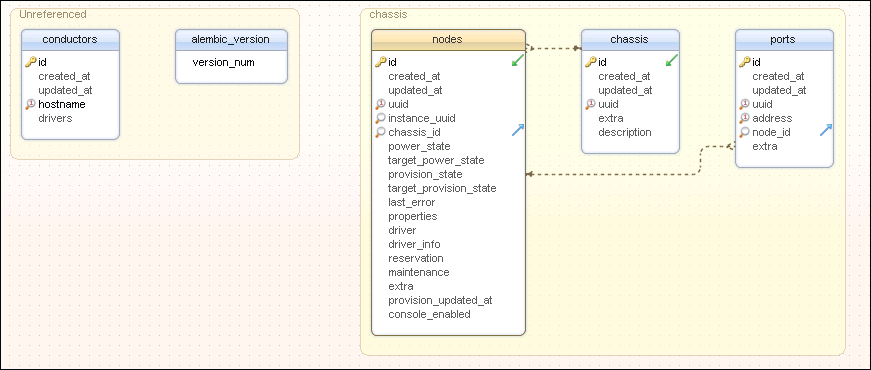
* PXE
* DHCP
* NBP
* TFTP
* IPMI
  1. 部署架构

云管理员使用RESTful API 注册 node 信息，比如 mac 和 ipmi 的信息，ironic-api 可以多实例。

安全考虑，ironic-conductor 最好安装在单独的节点上，唯一可连通数据面和IPMI管理面的服务。配置多实例，一为支持不同的 driver，二为管理 HA。每个实例需部署在不同的节点上，配置不同的 driver 以支持异构的硬件。



* 1. 数据架构



解释：

1. conductor : 翻译领导者，此处指的是 baremetal host，主要的作用是记录发现的drivers



1. alembic\_version
2. nodes : baremetal nodes
3. chassis :　便于管理 node 的概念
4. ports : baremetal nodes 上的所有的网口信息，address 为 MAC
   1. 代码架构

# 流程

* 1. baremetal instance deploy

前提：

1. 在计算节点上配置 tftp-server/ipmi/syslinux;
2. 创建适合可用硬件的 flavor，供 nova 使用；
3. Glance 需要如下镜像：

bm-deploy-kernel

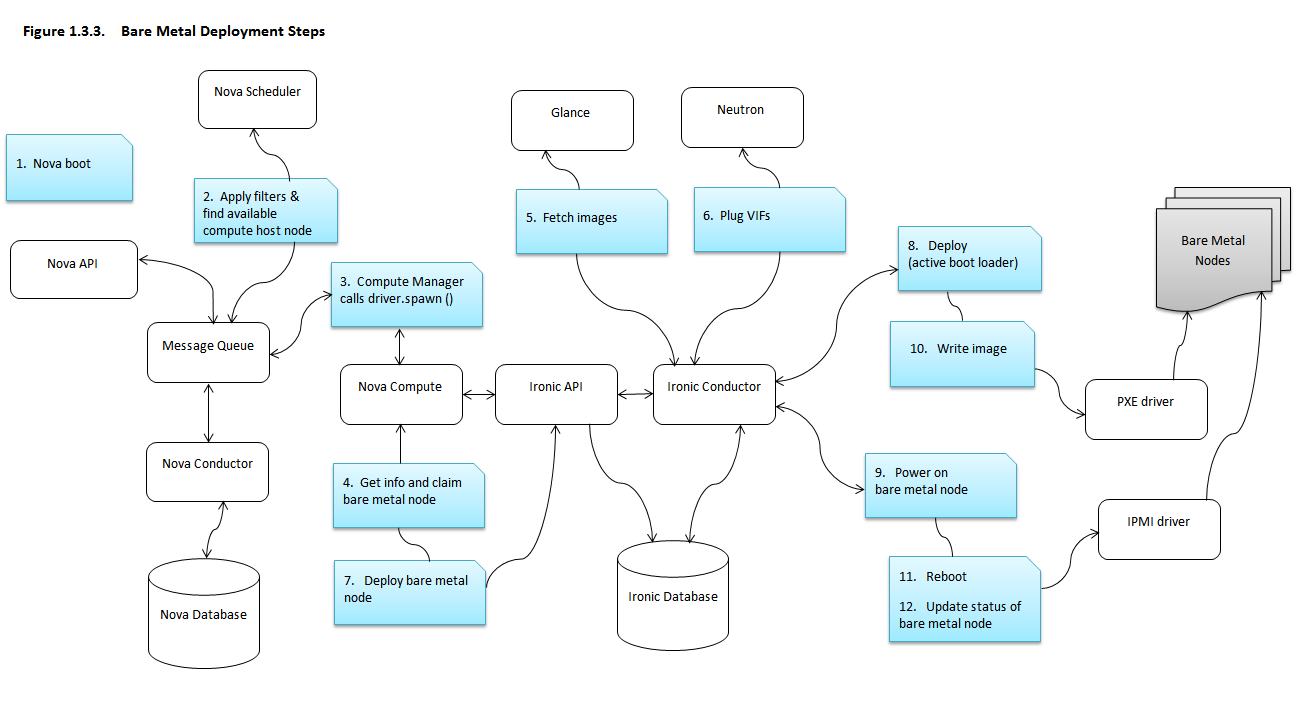
bm-deploy-ramdisk

user-image

user-image-vmlinuz

user-image-initrd

1. 注册为 ironic 的node；



1. A boot instance request comes in via the Nova API, through the message queue to the Nova scheduler.
2. Nova scheduler applies filter and finds the eligible compute node. Nova scheduler uses flavor extra\_specs detail such as ‘cpu\_arch’, ‘baremetal:deploy\_kernel\_id’, ‘baremetal:deploy\_ramdisk\_id’ etc to match the target physical node.
3. A spawn task is placed by the driver which contains all information such as which image to boot from etc. It invokes the driver.spawn from the virt layer of Nova compute.
4. Information about the bare metal node is retrieved from the bare metal database and the node is reserved.
5. Images from Glance are pulled down to the local disk of the Ironic conductor servicing the bare metal node.
6. Virtual interfaces are plugged in and Neutron API updates DHCP port to support PXE/TFTP options.
7. Nova’s ironic driver issues a deploy request via the Ironic API to the Ironic conductor servicing the bare metal node.
8. PXE driver prepares tftp bootloader.
9. The IPMI driver issues command to enable network boot of a node and power it on.
10. The DHCP boots the deploy ramdisk. The PXE driver actually copies the image over iSCSI to the physical node. It connects to the iSCSI end point, partitions volume, “dd” the image and closes the iSCSI connection. The deployment is done. The Ironic conductor will switch pxe config to service mode and notify ramdisk agent on the successful deployment.
11. The IPMI driver reboots the bare metal node. Note that there are 2 power cycles during bare metal deployment; the first time when powered-on, the images get deployed as mentioned in step 9. The second time as in this case, after the images are deployed, the node is powered up.
12. The bare metal node status is updated and the node instance is made available.
    1. baremetal instance delete



1. 使用 nova delete 删除 baremetal instance。
2. Nova api 发消息给 Nova compute，后者调用 destroy() 接口。
3. Nova Compute 调用 Ironic API 接口，后者发消息给 Ironic Conductor，触发 baremetal node tear down 流程。
4. 清除缓存的镜像，包括 deploy kernel 和 ramdisk、instance 使用的镜像。
5. 调用 ipmitool driver 将 node 下电。
6. 清除掉 Ironic DB node 记录中的 instance\_uuid 和 image info。
7. 清除掉 Ironic DB node 记录中的 网络信息。
   1. baremetal instance power on/off



1. 调用 Nova API 执行 start/stop。
2. Nova API 发消息给 Nova Compute，后者调用驱动的 poweron()/poweroff()。
3. Nova Compute 调用 Ironic API，后者发消息给 Ironic Conductor，触发 power state change 流程。
4. 调用 ipmitool driver 执行 baremetal instance 的 上下电，并更新状态到 Ironic 数据库。

# 参考文献

1. <http://docs.openstack.org/developer/ironic/>
2. <http://wiki.openstack.org/ironic>
3. <https://wiki.openstack.org/wiki/Baremetal>