# heat介绍

## heat介绍

### heat简介

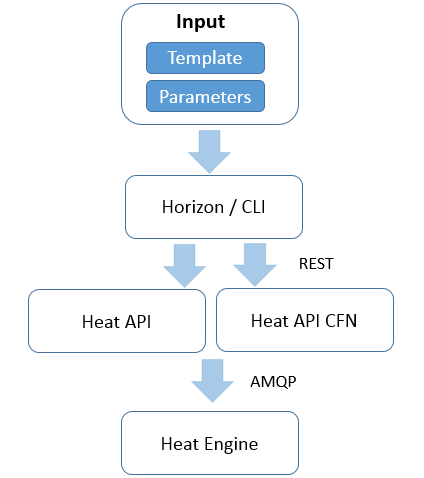
heat 是一个基于模板来编排复合云应用的服务。 它目前支持亚马逊的 CloudFormation 模板格式，也支持 Heat 自有的 Hot 模板格式。模板的使用简化了复杂基础设施，服务和应用的定义和部署。模板支持丰富的资源类型，不仅覆盖了常用的基础架构，**包括计算、网络、存储、镜像，还覆盖了像 Ceilometer 的警报、Sahara 的集群、Trove 的实例等高级资源**。

### heat架构

**heat-api** 组件实现 OpenStack 天然支持的 REST API。该组件通过把 API 请求经由 AMQP 传送给 Heat engine 来处理 API 请求。

**heat-api-cfn** 组件提供兼容 AWS CloudFormation 的 API，同时也会把 API 请求通过 AMQP 转发给 heat engine。

**heat-engine** 组件提供 Heat 最主要的协作功能。

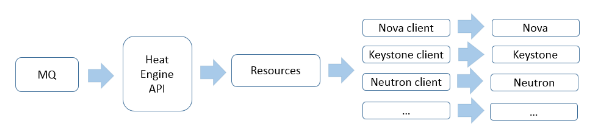


Heat架构图

用户在 Horizon 中或者命令行中提交包含模板和参数输入的请求，horizon 或者命令行工具会把请求转化为 REST 格式的 API 调用，然后调用 heat-api 或者是 heat-api-cfn。heat-api 和 heat-api-cfn 会验证模板的正确性，然后通过 AMQP 异步传递给 heat Engine 来处理请求。

当 heat Engine 拿到请求后，会把请求解析为各种类型的资源，每种资源都对应 OpenStack 其它的服务客户端，然后通过发送 REST 的请求给其它服务。通过如此的解析和协作，最终完成请求的处理。

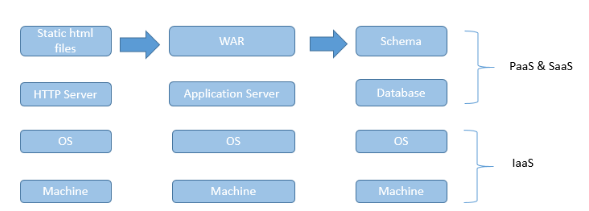
**heat Engine** 在这里的作用分为三层： 第一层处理 heat 层面的请求，就是根据模板和输入参数来创建 Stack，这里的 Stack 是由各种资源组合而成。 第二层解析 Stack 里各种资源的依赖关系，Stack 和嵌套 Stack 的关系。第三层就是根据解析出来的关系，依次调用各种服务客户段来创建各种资源。



heat Engine 结构

### 编排

编排，就是按照一定的目的依次排列。在 IT 的世界里，一个完整的编排一般包括**设置服务器上机器**、**安装 CPU**、**内存**、**硬盘**、**通电**、**插入网络接口**、**安装操作系统**、**配置操作系统**、**安装中间件**、**配置中间件**、**安装应用程序**、**配置应用发布程序**。对于复杂的需要部署在多台服务器上的应用，需要重复这个过程，而且需要协调各个应用模块的配置，比如配置前面的应用服务器连上后面的数据库服务器。下图显示了一个典型应用需要编排的项目。

编排

在云计算的世界里，机器这层就变为了虚拟的 VM 或者是容器。管理 VM 所需要的各个资源要素和操作系统本身就成了 IaaS 这层编排的重点。操作系统本身安装完后的配置也是 IaaS 编排所覆盖的范围。除此之外，提供能够接入 PaaS 和 SaaS 编排的框架也是 IaaS 编排的范围。

## heat使用

### 2.1服务搭建

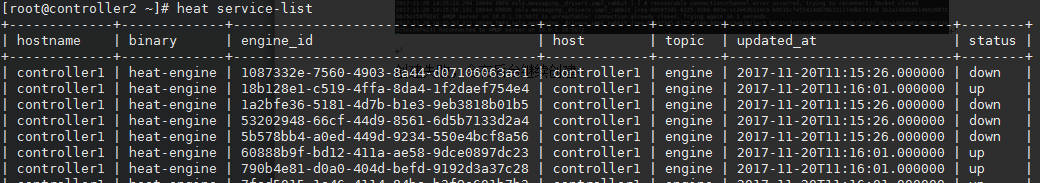
1、搭建参照官方（基于openstack基础组件之上）

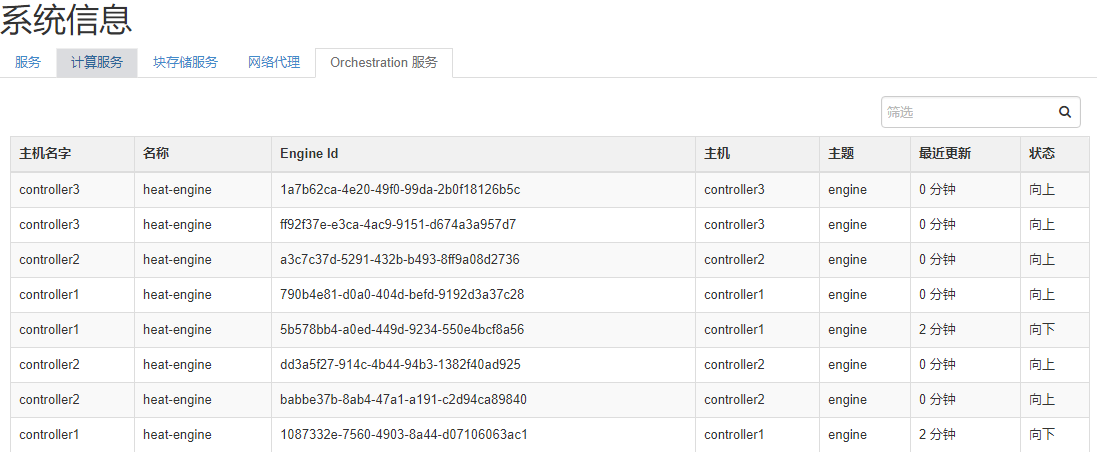
openstack-heat-api openstack-heat-api-cfn openstack-heat-engine三服务

2、openstack-ansible可直接使用。

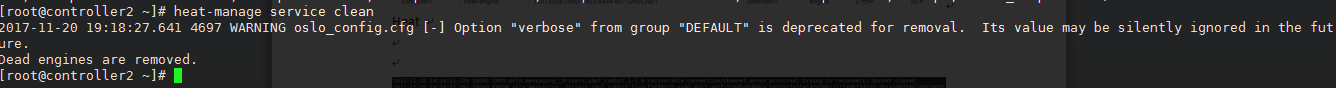
3、异常处理：

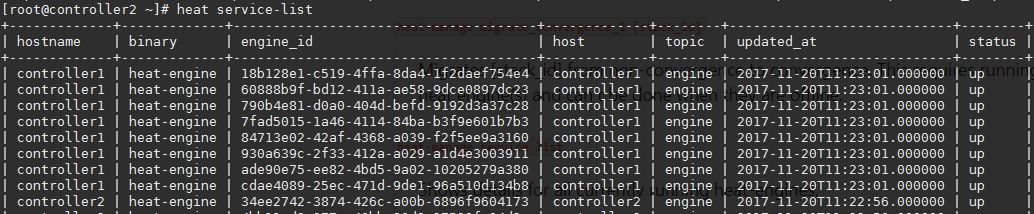
问题一：重启服务，处理api请求的heat-engine引擎重新生成，但依然留有上一次的引擎记录，且状态标记是down：





# **heat-manage service clean** #Clean dead engine records.





标记为down的引擎记录被清理了。

### 2.2 使用方法

**heat模板**

Heat 目前支持两种格式的模板，一种是基于 JSON 格式的 CFN 模板；另外一种是基于 YAML 格式的 HOT 模板。CFN 模板主要是为了保持对 AWS 的兼容性。HOT 模板是 Heat 自有的，资源类型更加丰富，更能体现出 Heat 特点的模板。

一个典型的 HOT 模板由下列元素构成：

**模板版本**：必填字段，指定所对应的模板版本，Heat 会根据版本进行检验。

**参数列表**：选填，指输入参数列表。

**资源列表**：必填，指生成的 Stack 所包含的各种资源。可以定义资源间的依赖关系，比如说生成 Port，然后再用 port 来生成 VM。

**输出列表**：选填，指生成的 Stack 暴露出来的信息，可以用来给用户使用，也可以用来作为输入提供给其它的 Stack。

对于 CFN 模板和 HOT 模板的不同，包括所支持的资源类型的不同，不在本文的讨论范围内。本文会主要用 HOT 模板。HOT 模板的全称是 Heat Orchestration Template，是 Heat 发展的重心。

heat\_template\_version: 2016-10-14

description:

*# a description of the template*

parameter\_groups:

*# a declaration of input parameter groups and order*

parameters:

*# declaration of input parameters*

resources:

*# declaration of template resources*

outputs:

*# declaration of output parameters*

conditions:

*# declaration of conditions*

heat\_template\_version的值不仅表示模板的格式，还表示将被验证和支持的特性。从Newton版本开始，版本可以是Heat版本的日期或Heat版本的代码名称。

**Conditions**

This optional section includes statements which can be used to restrict when a resource is created or when a property is defined. They can be associated with resources and resource properties in the resources section, also can be associated with outputs in theoutputs sections of a template.

**Note**: Support for this section is added in the Newton version.

翻译：**这个可选的部分包括可用于限制何时创建资源或何时定义属性的语句**。它们可以与资源部分中的资源和资源属性相关联，也可以与模板的输出部分中的输出相关联。

参考文档：

https://docs.openstack.org/heat/latest/template\_guide/hot\_spec.html#hot-spec

## 存在问题