Magnum中kubernetes temples结构分析

这里介绍magnum中kubernetes的templates的组织结构。

Kubernetes的templates分为三种类型：kubecluster、kubecluster\_coreos、kubecluster\_fedora\_ironic。分别用于启动三种不同的kubernetes。

1. Kubecluster.yaml

Kubecluster.yaml将启动一个具有一个或者多个node的kubenetes集群。它依赖当前目录下的两个yaml文件：kubemaster.yaml和kubeminion.yaml.

Kubecluster.yaml文件分为三个部分，parameters、resources、outputs三部分组成，parameters是一些环境参数的设置；resources主要是一些资源的配置；outputs是一些输出量。

1. Resources
   1. network resources：这个字段用于分配网络和路由。
   2. load balancer：负载均衡
   3. kubermaster：用于创建含有N个master的群组。
   4. kuberminion：用于创建含有N个minion的群组。
2. outputs
   1. api\_server\_address:用于从集群外部访问apiserver.
   2. Kube\_master:这是一个master IP地址列表，用于从外部用ssh访问master机器。
   3. Kube\_minions:所有minions的私有IP列表。
   4. Kube\_minions\_external:所有的minions的共有IP地址，用于外部访问查看log等。
3. kubecluster-coreos.yaml

这个模板将启动一个coreos的集群，依赖一下文件：

* 1. write-heat-params-master.yaml
  2. configure-kubernetes-master.yaml
  3. write-flannel-config.sh
  4. flannel-config.service.yaml
  5. enable-service-master.sh
  6. kube-user.yaml
  7. kube-examples.yaml
  8. cfn-signal.sh
  9. diable-selinux.sh
  10. coreos.params.yaml
  11. kubeminion-coreos.yaml

Resources：

1. Master\_wait设置。
2. Network resources：配置网络参数，一般会优于当前网络设置，提供了更详细的参数。
3. Security groups：提供多种网络通信。
4. Software config：将各个组件组织成一个多组件的用户数据档案。
5. kubecluster-fedora-ironic.yaml

这个模板将启动一个fedora的集群，依赖一下文件：

1. disable-selinux.sh
2. write-heat-params-master.yaml
3. configure-kubernetes-master.sh
4. enable-etcd.sh
5. configure-flannel.sh
6. enable-service-master.sh
7. kubenode-fedora-ironic.yaml

Resources：

1. master\_wait参数。
2. software config:设置需要运行的软件。
3. database server：设置k8s server的数据库。
4. kubemaster.yaml和kubemonion.yaml

这是内嵌的文件，用于定义单个k8s master和minion节点，它需要被父模板——kubecluster.yaml包含和调用。

Kubemaster.yaml中用到的其他文件：

1. write-heat-params-master.yaml
2. configure-etcd.sh
3. write-flannel-config.sh
4. flannel-config.service.yaml
5. enable-service-master.sh
6. kube-user.yaml
7. kube-example.yaml
8. disable-selinux.sh

kubeminion.yaml中用到的其他文件:

1. write-heat-params.yaml
2. configure-docker-storage.sh
3. kube-register.yaml
4. configure-kubenetes-minion.sh
5. kube-user.yaml
6. kube-examples.yaml
7. docker.service.yaml
8. enable-services-minion.sh
9. disable-selinux.sh
10. kubeminion-coreos.yaml和kubeminion-fedora-ironic.yaml

这两个分别是coreos和fedora两个cluster.yaml的子文件，用于补充两个父模板的设置。

Kubeminion-coreos.yaml用到的文件：

1. write-heat-params.yaml
2. coreos.params.yaml
3. add-to-docker-group.sh
4. configure-docker-storage.sh
5. kube-register.yaml
6. configure-kubernetes-minion.sh
7. kube-user.sh
8. kube-examples.yaml
9. docker.services.yaml
10. enable-service-minion.sh
11. cfn-signal.sh
12. disable-selinux.sh

kubeminion-fedora-ironic.yaml用到的文件：

1. write-heat-params.yaml
2. kube-register.yaml
3. configure-kubernetes-minions.sh
4. kube-user.yaml
5. kube-examples.yaml
6. docker.service.yaml
7. enable-services-minion.sh
8. disable-selinux.sh