

consul-template是consul公司开发的一款监控consul服务变化并自动生成配置文件的工具。生成配置文件后，也可以执行指定的一些命令。随着应用微服务化越来越流行，不少公司都开始使用consul了，当consul的监控的服务发生变化时，我们可能会自动重启nginx或者redis等服务。consul-template就是应用在此场景中。

安装

为了运行consul-template，需要先启动consul服务。consul和consul-template均是采用go语言编写的，可以到[consul官网](#)以及[consul-template官网](#)下载二进制包。

运行

启动consul

```
consul agent -dev #加上-dev参数,consul可以当做master节点运行
```

启动consul-template

首先我们需要编写一个模板文件(in.tpl):

```
{{ key "foo" }}
```

然后执行启动命令:

```
consul-template -template "in.tpl:out.txt" -once #添加-once参数后,consul-template运行一次就会退出
```

接着我们向consul kv存储中添加一对键值:

```
consul kv put foo bar
```

最后，我们可以看到当前目录下生成了一个out.txt文件:

```
bar
```

配置

上面的例子是官方提供的，可以供测试使用，实际使用的时候大多会通过指定配置文件的方式启动

```
consul-template -config "/my/config.hcl"
```

consul-template的不少配置项需要注意一下

```
#config.hcl
consul {
  #consul集群相关配置
  auth {
    enabled = true
    username = "test"
    password = "test"
  }

  address = "127.0.0.1:8500"
  token = "abcd1234"
  #当consul集群返回异常信息时,consul-template不会挂掉,而是会不断重试
  retry {
    enabled = true
    attempts = 12
    backoff = "250ms"
    max_backoff = "1m"
  }

  #通常情况下,consul-template会从consul集群的leader节点拉取配置,
  #当consul特别繁忙时,我们可以选择从follower节点拉取数据,此配置代表最大容忍的数据陈旧时间
  max_stale = "10m"

  #为了防止配置经常修改导致模板不断被渲染,影响系统的稳定
  #我们可以设置此参数,此参数控制模板两次渲染之间的间隔时间
  wait {
    min = "5s"
    max = "10s"
  }

  #下面这个参数很重要。consul-template从consul中取数据时采用的是阻塞的方法
  # (consul-template设置的默认超时时间为1min,并且不能修改),
  #假设一个模板依赖50个资源,一共有50个consul-template实例的话,就会有2500个阻塞请求,
  #会对consul造成比较大的压力
  #打开deduplicate参数的话,consul-template会自动组成一个集群,leader节点会监控所有的
  资源,并将整合后的资源再存储到consul中
  #其他consul-template节点对于这个模板只用监控这一个key即可
  #不过在0.18.5版本,此处有bug,具体可以看https://github.com/hashicorp/consul-template/blob/master/CHANGELOG.md
  deduplicate {
    enabled = true
    prefix = "consul-template/dedup/"
  }

  # 配置模板
  template {
    source = "/path/on/disk/to/template.ctmpl"
    destination = "/path/on/disk/where/template/will/render.txt"
    create_dest_dirs = true
  }
}
```

```
contents = "{{ keyOrDefault \"service/redis/maxconns@east-aws\" \"5\" }}"
command = "restart service foo"
command_timeout = "60s"
error_on_missing_key = false
perms = 0600
backup = true
left_delimiter = "{{"
right_delimiter = "}}"
wait {
  min = "2s"
  max = "10s"
}
```

模板语法

consul-template的语法其实采用是go本身的模板语法，在此基础上添加了不少自定义函数

```
{{ key "service/redis/maxconns" }}
```

key其实是consul-template提供的一个函数，支持从consul中取值。

同时consul-template也提供了诸如**datacenters** **file** **keyExists** **keyOrDefault**等函数，具体可以参考官方文档<https://github.com/hashicorp/consul-template#templating-language>