Mesos DNS服务发现

当同时运行应用程序多个副本时，必须能够发现它们在哪里运行，并能够连接它们。这在出现故障时尤其重要，这时Mesos会在其他集群上重启服务，但是客户端需要重新连接上新启动的服务，发现服务的两种常用做法：

1) 基于域名服务

2) 专业的服务发现方案（Zookeeper、Consul、Serf、SkyDNS等）

基于域名服务（DNS）的服务发现，很难提供服务相关的丰富信息，但是专业方案能够提供服务健康和之外的更多信息。

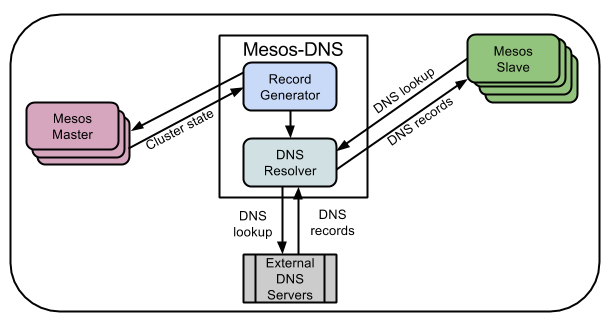
Mesos-DNS是针对Mesos的基于DNS的服务器发现机制，将分配给Mesos应用程序的名称解释为这些服务运行的IP地址和端口。其接口非常简单，接收DNS请求、回复DNS记录，Framework只需使用标准的DNS查询就可以通过名称解析出地址，当集群状态改变时，可以很容易地切换到其他机制来更新DNS记录，而无须改变框架本身。

目前，Mesos-DNS只支持ANY/A和SRV DNS记录，对于Mesos域里的其他类型记录，会返回NXDOMAIN，也就是它不能用于需要PTR记录的反向寻找。A记录将主机名和IP地址关联起来，Mesos-DNS为task.framework.domain主机名生成一个A记录，该主机名提供指定的slave的IP地址，运行框架启动的任务。比如，其他Mesos任务可以通过db.marathon.mesos来为Marathon框架启动的db服务发现IP地址。另外，只会为Mesos分配端口的任务生成SRV记录，为防框架启动了多个同名服务，Mesos-DNS会返回多个DNS记录，每个任务一条记录，记录顺序是随机的。这提供了复杂的负载均衡能力。

Mesos-DNS没有提供其他功能，比如健康检查或者应用的生命周期管理。这有助于保持Mesos-DNS架构的简单和无状态。它不使用持久化存储、复制和一致性策略。无状态架构使得在大规模Mesos集群里非常易于扩展。该设计的后果之一就是Mesos-DNS本身无法容错，必须依赖于外部实体。Mesos-DNS在Mesos master不可用时还能继续为现有服务提供stale DNS记录。注意，Mesos-DNS能够处理Mesos master故障转移，因为它会连接到新的Mesos master上。

# 1.Mesos-DNS

框架如下图所示：



Mesos-DNS有两个主要组件：

1）DNS记录生成器：负责为所有运行着的应用程序生成DNS A和SRV记录，它周期性的查询master，得到所有运行的应用程序及每个应用运行的任务信息。因此，记录生成器知道所有任务的状态，比如启动、完成、失败或重启。记录生成器以DNS记录的格式保存各种服务的最新状态

2）DNS解析器，负责处理DNS请求并恢复请求，解析器直接回复Mesos上运行着的任务，为外部请求选择随机的DNS服务器，就像普通DNS服务器的行为一样。

# 2.安装和使用

1）获取执行文件

mkdir /usr/local/mesos-dns/

wget -O /usr/local/mesos-dns/mesos-dns https://github.com/mesosphere/mesos-dns/releases/download/v0.5.1/mesos-dns-v0.5.1-linux-amd64

2）配置参数

{

"zk": "zk://fys1.cmss.com:2181/mesos", //mesos master的zk path

"refreshSeconds": 60,

"ttl": 60,

"domain": "mesos",

"port": 53,

"resolvers": ["10.133.47.16","112.4.0.55"], //DNS服务器

"timeout": 5,

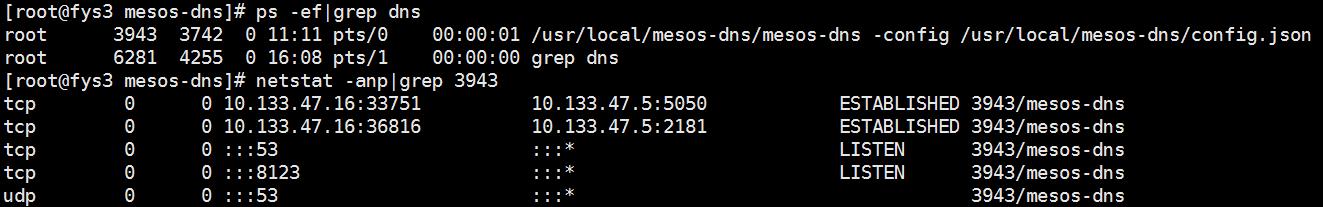
"email": "root.mesos-dns.mesos"

}

3）启动命令

/usr/local/mesos-dns/mesos-dns -config /usr/local/mesos-dns/config.json

结果如下：



其中8123端口是HTTP端口

4）使用marathon创建测试用例Tomcat

{

"id": "tomcat",

"cmd": "mv \*.war apache-tomcat-\*/webapps && cd apache-tomcat-\* && sed \"s/8080/$PORT/g\" < ./conf/server.xml > ./conf/server-mesos.xml &&chmod 755 ./bin/catalina.sh && ./bin/catalina.sh run -config ./conf/server-mesos.xml",

"mem": 512,

"cpus": 1.0,

"instances": 1,

"uris": [

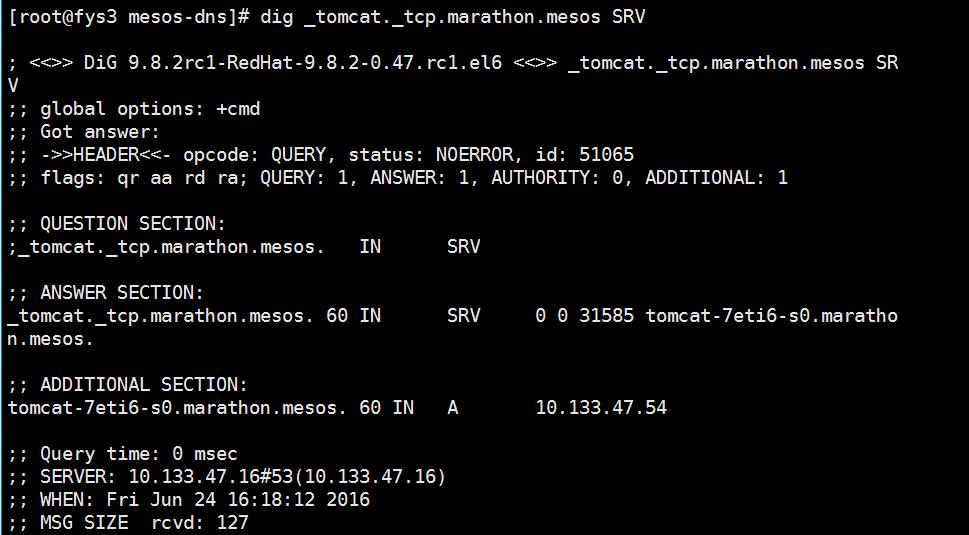
"http://10.133.17.27/software/tomcat/tarball/apache-tomcat-7.0.62.zip",

"http://10.133.17.27/tmp/test.war"

]

}

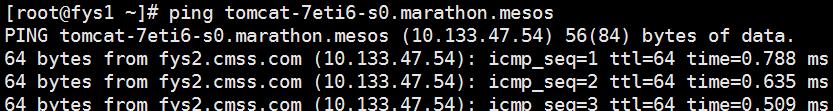
通过dig查看如下：



测试：

ping tomcat-7eti6-s0.marathon.mesos

结果如下:



HTTP 查看结果如下：



# 3.深入解析

## 3.1 Marathon config

Mesos-DNS在启动使用json格式的文件，通过-config=pathto/file.json来指定。config文件包括以下字段：

{

"zk": "zk://10.101.160.15:2181/mesos", *//mesos集群的zk path*

"masters": ["10.101.160.15:5050", "10.101.160.16:5050", "10.101.160.17:5050"],

*//mesos master，Mesos-DNS自动查找leading master,然后查询task的执行状态， //zk或者mastes字段(都是static的数据)，只需要提供一个*

"refreshSeconds": 60, *//mesos-dns数据更新周期（请求mesos-master）*

"ttl": 60,  *//DNS记录有效时间，默认300秒*

"domain": "mesos", *// Mesos 集群的domain name*

"port": 53,  *//Mesos-DNS的使用的端口，默认53*

"resolvers": ["169.254.169.254"], *//Mesos-DNS使用的外部DNS来解析非mesos domain*  *的DNS请求，可以使用8.8.8.8这种常用DNS Server*

"timeout": 5, *//请求的timeout时间*

"httpon": true, *//是否相应HTTP请求*

"dnson": true, *//Mesos-DNS是否监听DNS请求*

"httpport": 8123, *//HTTP Port*

"externalon": true, *//Mesos-DNS是否响应非Mesos-Dns请求*

"listener": "10.101.160.16", *//Mesos-DNS的ip地址*

"SOAMname": "ns1.mesos", *//Name Server的domain name*

"SOARname": "root.ns1.mesos", *// Mainbox*

"SOARefresh": 60,

"SOARetry": 600,

"SOAExpire": 86400,

"SOAMinttl": 60,

"IPSources": ["netinfo", "mesos", "host"]

}

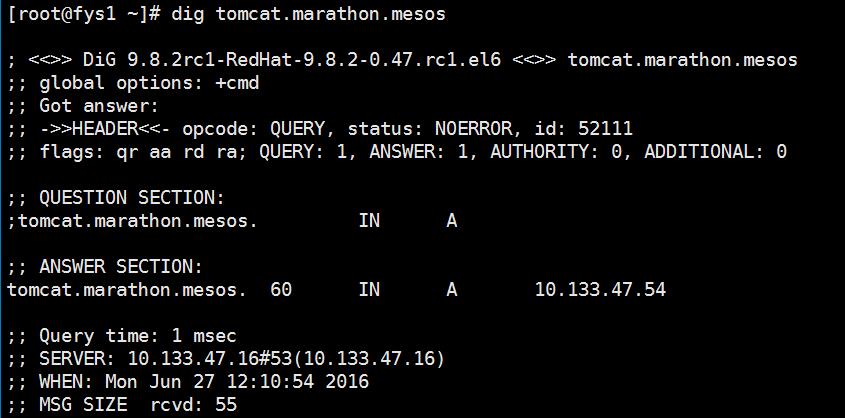
## 3.2 Service Naming

Mesos-DNS为Mesos tasks定义了DNS Domain，默认为.mesos。通过通过Mesos-DNS查看到运行tasks的Mesos Domain中的A/SRV记录。

1）A 记录

A record形式为hostname -> ip address的记录。通过mesos framework启动的task，Mesos-DNS会产生task.framework.domain形式的A record。

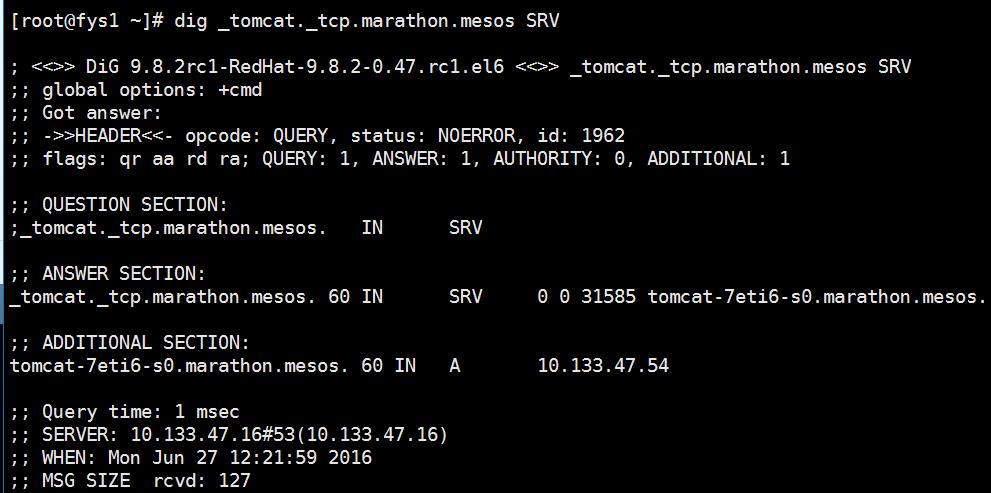
例如，其他的Mesos Task通过查询tomcat.marathon.mesos来获取tomcat service的ip



tomcat.marathon.mesos启动在节点10.133.47.54上

2）SRV记录

SRV记录为service和hostname:ip的记录。Mesos-DNS为task生成一条SRV记录，格式为：\_task.\_protocol.framework.domain，protocol为tcp或者upd。例如mesos task为tomcat服务:



Mesos-DNS支持通过SRV Record产生task的discover信息。

| **Service** | **CT-IP Avail** | **DI Avail** | **Target Host** | **Target Port** | **A (Target Resolution)** |
| --- | --- | --- | --- | --- | --- |
| *{task}.*{proto}.framework.domain | no | no | {task}.framework.slave.domain | host-port | slave-ip |
|  | yes | no | {task}.framework.slave.domain | host-port | slave-ip |
|  | no | yes | {task}.framework.domain | di-port | slave-ip |
|  | yes | yes | {task}.framework.domain | di-port | container-ip |
| *{task}.*{proto}.framework.slave.domain | n/a | n/a | {task}.framework.slave.domain | host-port | slave-ip |

3）其他Records

Mesos-DNS也会生成以下special record：上

leading master: A Redcord(leading.domain)和SRV record(\_leader.\_tcp.domain和\_leader.\_udp.domain)。

framework schedulers:

A Record({framework}.domain)和SRV record(\_framework.\_tcp.{framework}.domain)

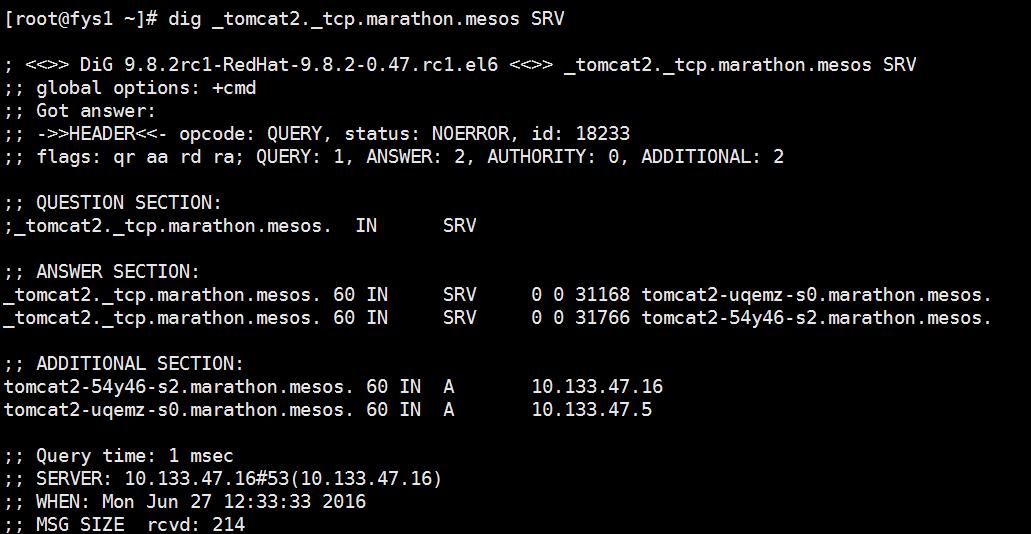
meoss maaster:

A record:master.domain, SRV Record:\_master.\_tcp.domain和\_master.\_upd.domain

slave-master:

A record: slave.domain,SRV Record: \_slave.\_tcp.domain

注：如果使用相同的名字启动多个task，则会返回多个record。



## 3.3 HTTP Interface

提供以下REST API(GET)

GET /v1/version，列举Mesos-DNS的版本

GET /v1/config,，列举Mesos-DNS的配置新

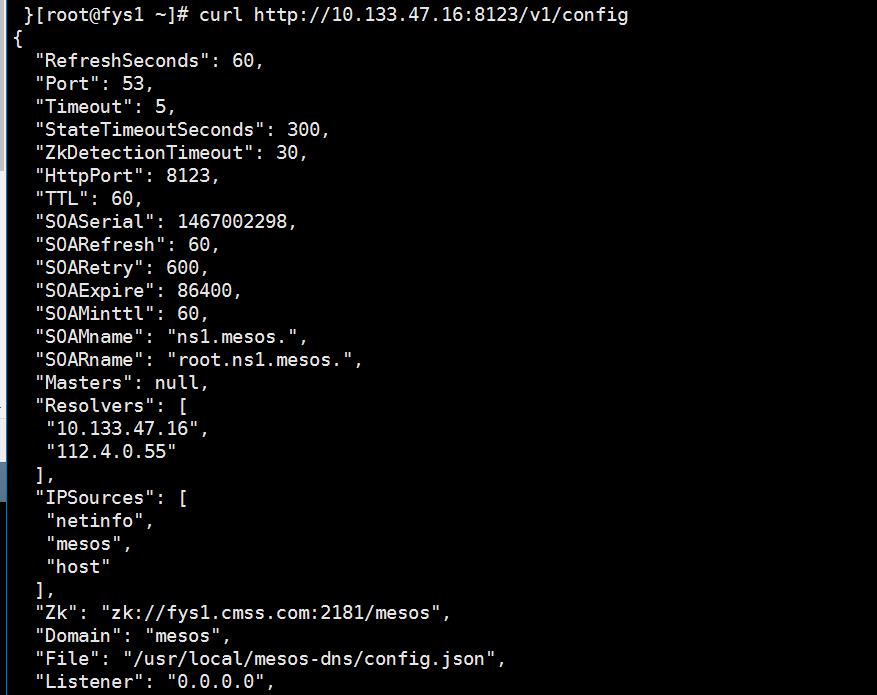
GET /v1/hosts/{host}，列举一个host的ip地址

GET /v1/service/{service}，列举一个host,ip地址和service的端口号

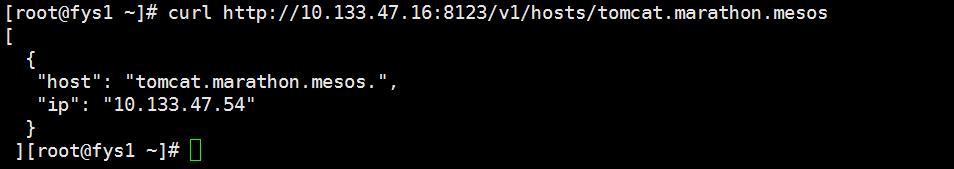
1）GET /v1/version



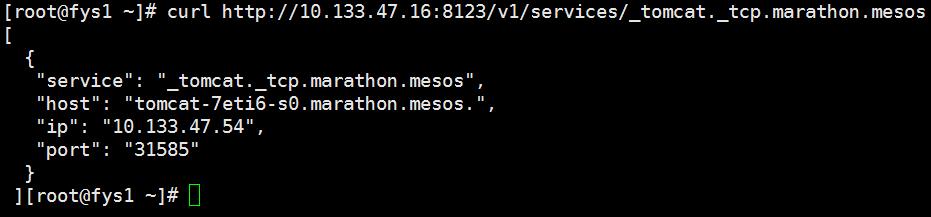
2）GET /v1/config



3）GET /v1/hosts/{host}

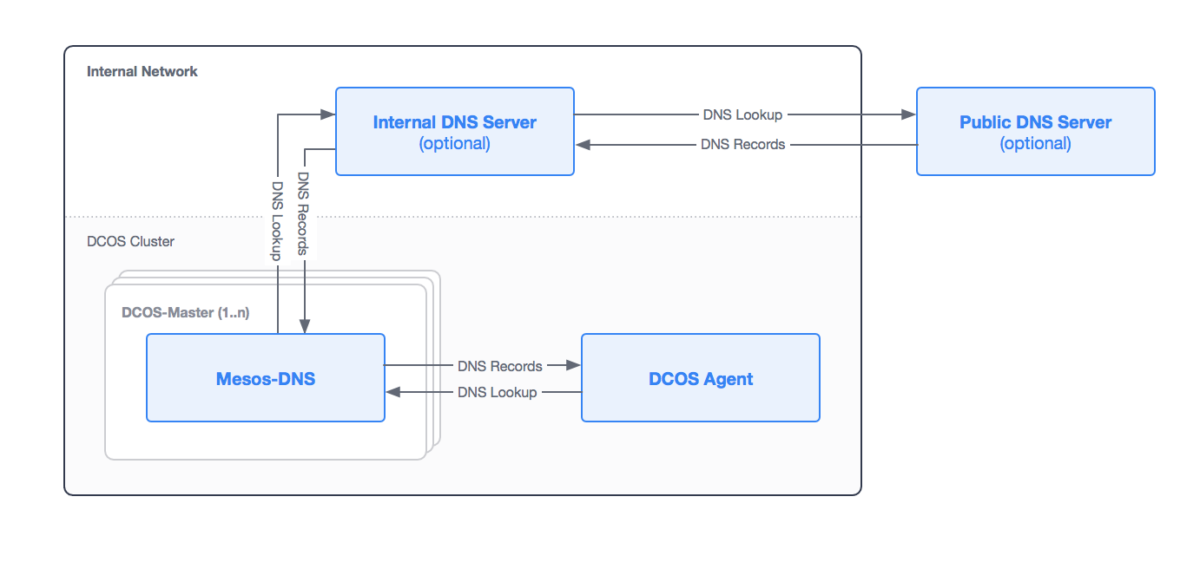


4）GET /v1/services/{service}



# 4.Mesos-DNS的PASS能力

在生产环境中，在Mesos Cluster中往往要将原dns server指向mesos-dns。但是往往希望集群外的集群也可以访问到mesos-dns，使用时要将dns server设置成mesos-dns(mesos-domain).



参考文献：

1）官方使用文档：

https://mesosphere.github.io/mesos-dns/docs/tutorial.html

2）github

https://github.com/mesosphere/mesos-dns

3）Marathon与Mesos-DNS的使用框架

http://programmableinfrastructure.com/guides/service-discovery/mesos-dns-haproxy-marathon/

4）命名规则：https://tools.ietf.org/html/rfc952

5) 社区源码：https://github.com/mesosphere/mesos-dns/releases

http://programmableinfrastructure.com/guides/service-discovery/mesos-dns-haproxy-marathon/  
https://mesosphere.github.io/mesos-dns/docs/

http://mesosphere.github.io/mesos-dns/docs/tutorial.html

http://mesosphere.github.io/mesos-dns/docs/tutorial-systemd.html