1.概念

Marathon可以通过Constraints来控制其app在何处运行

我们可以通过Marathon的[REST API](#)或者[Marathon gem](#)(marathon client)来设置app的constraints配置。

Constraints由三个部分组成：

<mark>字段名(field name)， 操作(operator), 可选参数(optional parameter)</mark>

其中字段名(field name)可以为mesos的一个slave的hostname或者Mesos slave的attribute。

2.实战

接下来我们结合实战，来介绍下Marathon Constrains的用法

Slave IP	Attribute
192.168.100.27	无
192.168.100.30	rack_id:rack-1
192.168.100.31	rack_id:rack-1
192.168.100.32	rack_id:rack-2

字段名

Hostname字段

hostname 匹配slave的hostnames。

hostname支持所有的operator

Attribute字段

attribute 匹配Mesos slave的attributes字段。我们可以通过在mesos节点上运行mesos-slave --help来学习如何设置mesos slave的attributes。
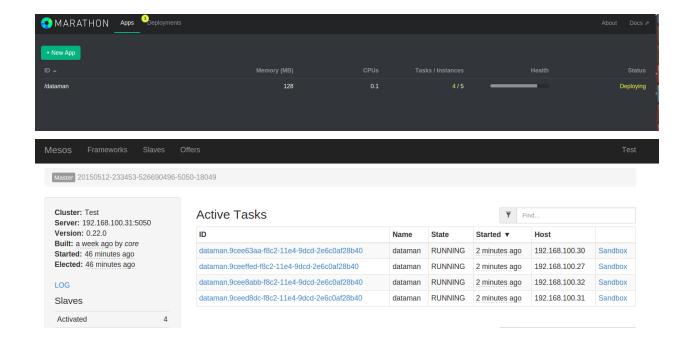
操作

UNIQUE operator

`UNIQUE` tells Marathon to enforce uniqueness of the attribute across all of an app's tasks.

```
$ curl -v -X POST http://192.168.100.30:8080/v2/apps \
    -H Content-Type:application/json -d '{
    "id":"dataman",
    "container": {
    "type": "DOCKER",
      "docker": {
        "image": "192.168.100.13:5000/fchen/mynginx",
        "network": "BRIDGE",
        "portMappings": [
          { "containerPort": 80, "hostPort": 0, "protocol": "tcp"}
        ]
      }
    },
    "constraints": [["hostname", "UNIQUE"]],
    "cmd": "/usr/sbin/nginx -c /etc/nginx/nginx.conf",
    "cpus": 0.1,
    "mem": 128.0,
    "instances": 5
  }'
```
field name = hostname, operator = UNIQUE, 在每个slave上只起一个instance。

由于我们申请了5个nginx instance，而我们Mesos集群节点只有4个slave，所以我们在Marathon的管理界面上看到app dataman为Deploying状态。
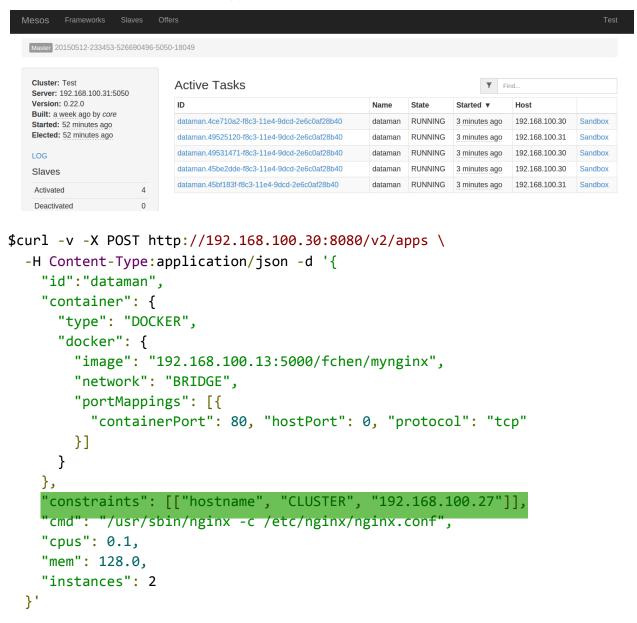
CLUSTER operator

CLUSTER allows you to run all of your app's tasks on slaves that share a certain attribute. This is useful for example if you have apps with special hardware needs, or if you want to run them on the same rack for low latency.

```
$ curl -v -X POST http://192.168.100.30:8080/v2/apps \
    -H Content-Type:application/json -d '{
      "id":"dataman",
      "container": {
        "type": "DOCKER",
        "docker": {
          "image": "192.168.100.13:5000/fchen/mynginx",
          "network": "BRIDGE",
          "portMappings": [
            { "containerPort": 80, "hostPort": 0, "protocol": "tcp"}
          ]
        }
      },
      "constraints": [["rack_id", "CLUSTER", "rack-1"]],
      "cmd": "/usr/sbin/nginx -c /etc/nginx/nginx.conf",
      "cpus": 0.1,
      "mem": 128.0,
      "instances": 5
```

```
    }'
```

CLUSTER允许我们将app发布到Mesos特定的slave节点上。如果你的app对物理机的硬件有要求，那么这个特性就很有用了。

我们可以看到marathon将我们的app都发布到了rack-1节点上去了。


```
$curl -v -X POST http://192.168.100.30:8080/v2/apps \
  -H Content-Type:application/json -d '{
    "id":"dataman",
    "container": {
      "type": "DOCKER",
      "docker": {
        "image": "192.168.100.13:5000/fchen/mynginx",
        "network": "BRIDGE",
        "portMappings": [{
          "containerPort": 80, "hostPort": 0, "protocol": "tcp"
        }]
      }
    },
    "constraints": [["hostname", "CLUSTER", "192.168.100.27"]],
    "cmd": "/usr/sbin/nginx -c /etc/nginx/nginx.conf",
    "cpus": 0.1,
    "mem": 128.0,
    "instances": 2
  }'
```

我们还可以指定field name为hostname，并将optional parameter指定为我们希望app运行的slave的hostname。

这样我们将2个dataman的instance发布到了192.168.100.27这台slave上。

GROUP_BY operator

GROUP_BY can be used to distribute tasks evenly across racks or datacenters for high availability.

```
$curl -v -X POST http://192.168.100.30:8080/v2/apps \
  -H Content-Type:application/json -d '{
    "id":"dataman",
    "container": {
      "type": "DOCKER",
      "docker": {
        "image": "192.168.100.13:5000/fchen/mynginx",
        "network": "BRIDGE",
        "portMappings": [{
          "containerPort": 80, "hostPort": 0, "protocol": "tcp"
        }]
      }
    },
    "constraints": [["rack_id", "GROUP_BY"]],
    "cmd": "/usr/sbin/nginx -c /etc/nginx/nginx.conf",
    "cpus": 0.1,
    "mem": 128.0,
    "instances": 6
}'
```
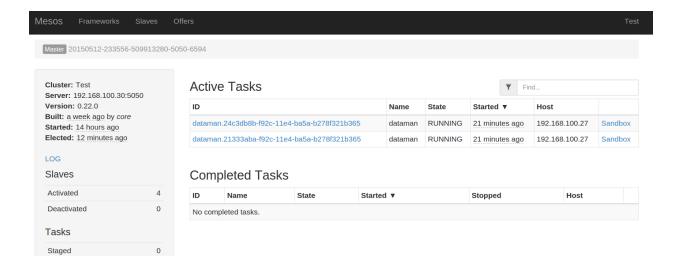
GROUP_BY帮我们将instance均匀的分布到不同的机架上，保证了高可用性。可以看到rack-1和rack-2上的instance个数均为3个。

Master 20150512-233556-509913280-5050-6594

Cluster: Test
Server: 192.168.100.30:5050
Version: 0.22.0
Built: a week ago by *core*
Started: 15 hours ago
Elected: an hour ago

LOG

Slaves

Activated	4
Deactivated	0

Active Tasks

Find...

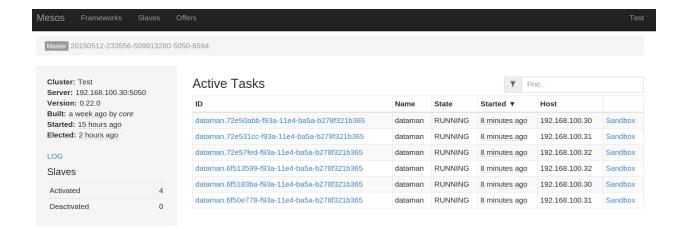| ID	Name	State	Started ▼	Host	
dataman.32c2c427-f937-11e4-ba5a-b278f321b365	dataman	RUNNING	14 minutes ago	192.168.100.32	Sandbox
dataman.2f2f1616-f937-11e4-ba5a-b278f321b365	dataman	RUNNING	14 minutes ago	192.168.100.30	Sandbox
dataman.2f2ec7f5-f937-11e4-ba5a-b278f321b365	dataman	RUNNING	14 minutes ago	192.168.100.32	Sandbox
dataman.2b9acbc4-f937-11e4-ba5a-b278f321b365	dataman	RUNNING	15 minutes ago	192.168.100.30	Sandbox
dataman.2b9a7da3-f937-11e4-ba5a-b278f321b365	dataman	RUNNING	15 minutes ago	192.168.100.32	Sandbox
dataman.2b9a5692-f937-11e4-ba5a-b278f321b365	dataman	RUNNING	15 minutes ago	192.168.100.31	Sandbox

LIKE operator

`LIKE` accepts a regular expression as parameter, and allows you to run your tasks only on the slaves whose field values match the regular expression.

```
curl -v -X POST http://192.168.100.30:8080/v2/apps \
  -H Content-Type:application/json -d '{
    "id":"dataman",
    "container": {
      "type": "DOCKER",
      "docker": {
        "image": "192.168.100.13:5000/fchen/mynginx",
        "network": "BRIDGE",
        "portMappings": [{
          "containerPort": 80, "hostPort": 0, "protocol": "tcp"
        }]
      }
    },
    "constraints": [["rack_id", "LIKE", "rack-[1-2]"]],
    "cmd": "/usr/sbin/nginx -c /etc/nginx/nginx.conf",
    "cpus": 0.1,
    "mem": 128.0,
    "instances": 6
}'
```

LIKE操作可以将我们的instance发布到我们希望的slave上，匹配正则表达式。

注意：LIKE操作必须带上parameter。

Cluster: Test
Server: 192.168.100.30:5050
Version: 0.22.0
Built: a week ago by *core*
Started: 15 hours ago
Elected: 2 hours ago

LOG

Slaves

Activated	4
Deactivated	0

Active Tasks

ID	Name	State	Started ▼	Host	
dataman.72e50abb-f93a-11e4-ba5a-b278f321b365	dataman	RUNNING	8 minutes ago	192.168.100.30	Sandbox
dataman.72e531cc-f93a-11e4-ba5a-b278f321b365	dataman	RUNNING	8 minutes ago	192.168.100.31	Sandbox
dataman.72e57fed-f93a-11e4-ba5a-b278f321b365	dataman	RUNNING	8 minutes ago	192.168.100.32	Sandbox
dataman.6f513599-f93a-11e4-ba5a-b278f321b365	dataman	RUNNING	8 minutes ago	192.168.100.32	Sandbox
dataman.6f5183ba-f93a-11e4-ba5a-b278f321b365	dataman	RUNNING	8 minutes ago	192.168.100.30	Sandbox
dataman.6f50e778-f93a-11e4-ba5a-b278f321b365	dataman	RUNNING	8 minutes ago	192.168.100.31	Sandbox

UNLIKE operator

Just like `LIKE` operator, but only run tasks on slaves whose field values don't match the regular expression.

```
curl -v -X POST http://192.168.100.30:8080/v2/apps \
  -H Content-Type:application/json -d '{
    "id":"dataman",
    "container": {
      "type": "DOCKER",
      "docker": {
        "image": "192.168.100.13:5000/fchen/mynginx",
        "network": "BRIDGE",
        "portMappings": [{
          "containerPort": 80, "hostPort": 0, "protocol": "tcp"
        }]
      }
    },
    "constraints": [["rack_id", "UNLIKE", "rack-[2,3]"]],
    "cmd": "/usr/sbin/nginx -c /etc/nginx/nginx.conf",
    "cpus": 0.1,
    "mem": 128.0,
    "instances": 6
}'
```

同LIKE操作，UNLIKE不将instance发布到匹配的slave上，参数匹配正则表达式。

Master 20150512-233556-509913280-5050-6594

Cluster: Test
Server: 192.168.100.30:5050
Version: 0.22.0
Built: a week ago by *core*
Started: 15 hours ago
Elected: 2 hours ago

LOG

Slaves

Activated	4
Deactivated	0

Active Tasks

[Find...]

ID	Name	State	Started ▼	Host	
dataman.56130ad3-f93c-11e4-ba5a-b278f321b365	dataman	RUNNING	just now	192.168.100.27	Sandbox
dataman.557a4d42-f93c-11e4-ba5a-b278f321b365	dataman	RUNNING	just now	192.168.100.30	Sandbox
dataman.557a2631-f93c-11e4-ba5a-b278f321b365	dataman	RUNNING	just now	192.168.100.31	Sandbox
dataman.527ec080-f93c-11e4-ba5a-b278f321b365	dataman	RUNNING	a minute ago	192.168.100.27	Sandbox
dataman.51e629fe-f93c-11e4-ba5a-b278f321b365	dataman	RUNNING	a minute ago	192.168.100.30	Sandbox
dataman.51e6c63f-f93c-11e4-ba5a-b278f321b365	dataman	RUNNING	a minute ago	192.168.100.31	Sandbox