基于docker搭建分布式系统的安装文档

说明：看下面文档之前先看http://www.jianshu.com/p/e26570d1b297?mType=Group

1、环境

win10、电脑内存大于4G

2、工具安装

git、maven、docker（docker安装过程中最好把git、virtualbox等选项都勾上，一起安装）

docker安装参考：http://www.jianshu.com/p/e26570d1b297?mType=Group

3、docker运行

3.1 docker安装完成后，桌面出现Docker Quickstart Terminal和Kitematic (Alpha)两个快捷方式

首先双击启动Docker Quickstart Terminal，启动完成后，终端会显示一个ip地址192.168.99.100，这个地址就是docker供外部访问的ip。

启动过程中可能会出现如下问题：

boot2docker.iso下载失败，所以需要翻墙下载，或者下载别人共享的镜像，注意版本需要和docker所需要的版本一致，不能下最新的也不能低于所需的版本。

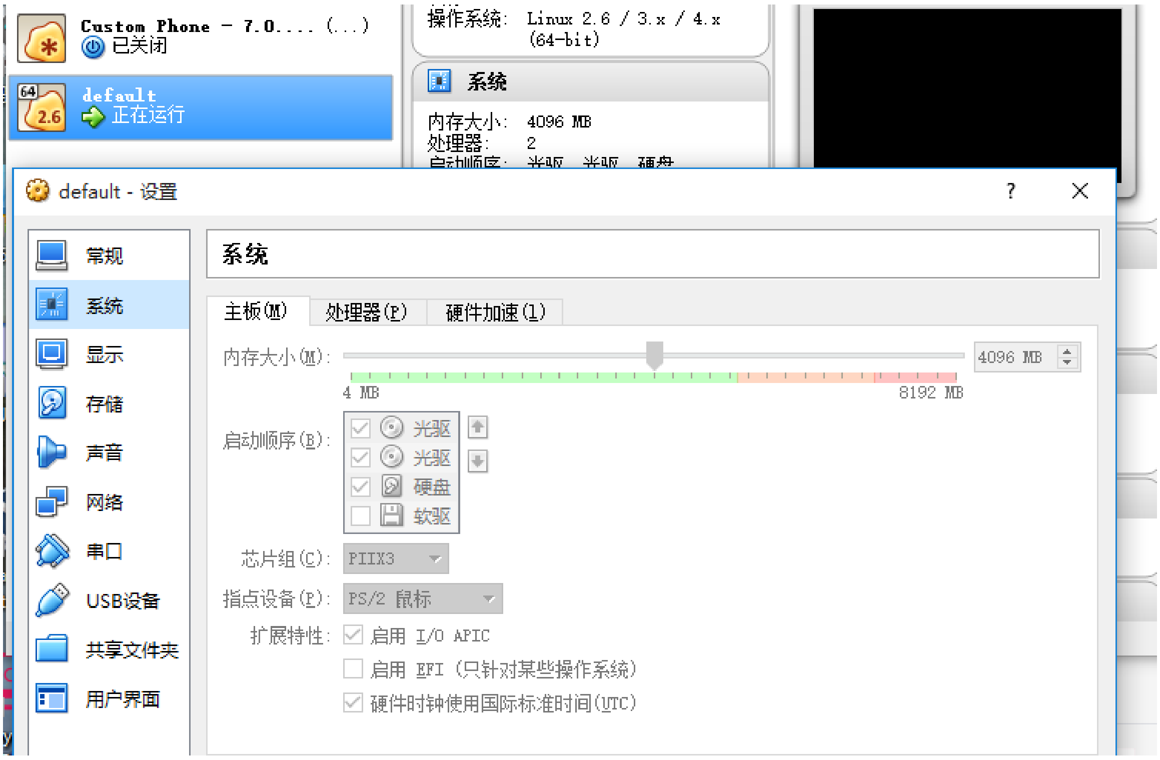
下载完后之后，放入C盘，我的路径是C:\Users\admin\.docker\machine\cache

首先情况cache，然后再放入。

参考网站： https://my.oschina.net/aixiaohua/blog/651171

3.2 修改VirtualBox中docker虚拟机的内存和处理器核数

基于PiggyMetrics搭起分布式系统会起10个容器，所需内存比较大，所以需要打开VirtualBox，修改docker对应的虚拟机的内存和处理器，默认名字是default



Custom Phone - 7.0.... (...)
已关闭

default
正在运行

操作系统: Linux 2.6 / 3.x / 4.x (64-bit)

系统

内存大小: 4096 MB

处理器: 2

启动顺序: 光驱 光驱 硬盘

default - 设置

系统

主板(M) 处理器(P) 硬件加速(L)

内存大小(M): 4096 MB

启动顺序(B): ☑ 光驱 ☑ 光驱 ☑ 硬盘 □ 软驱

芯片组(C): PIIX3

指点设备(P): PS/2 鼠标

扩展特性: ☑ 启用 I/O APIC

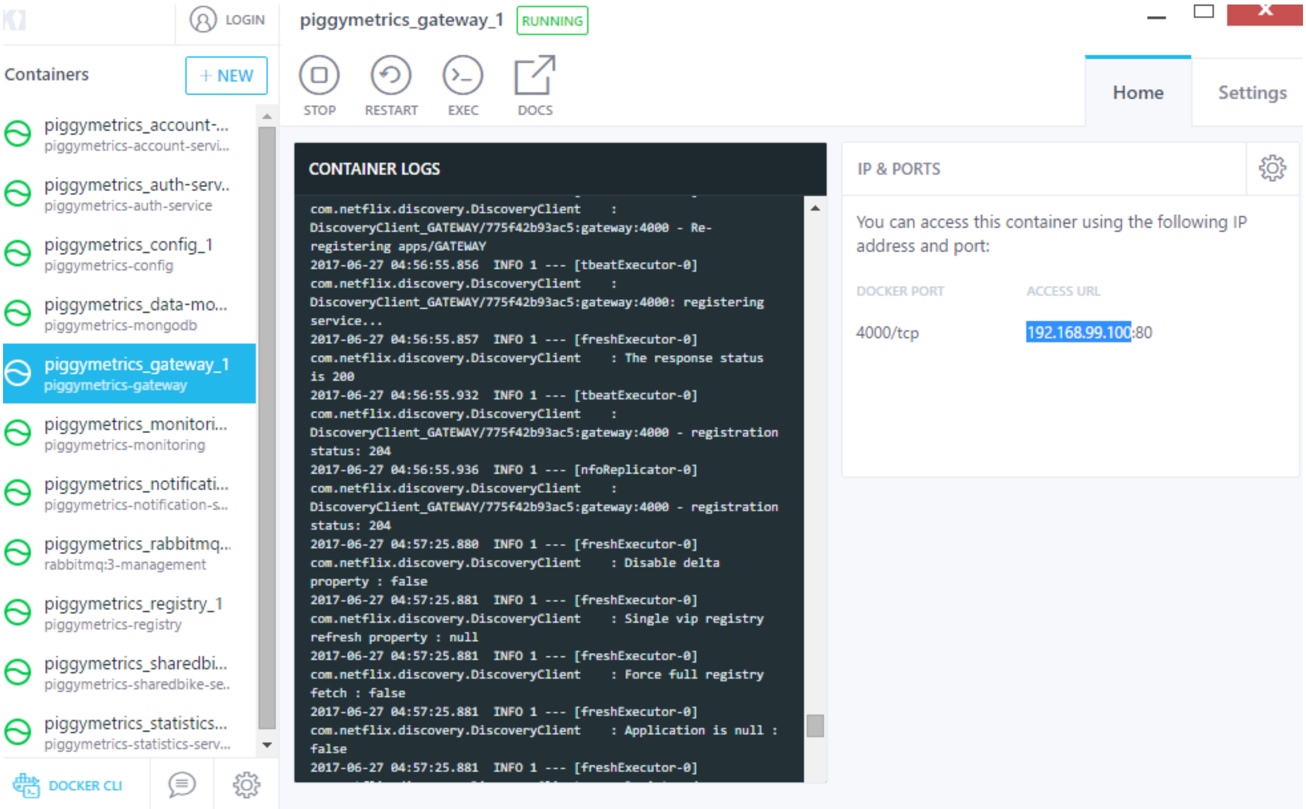
□ 启用 EFI (只针对某些操作系统)

☑ 硬件时钟使用国际标准时间(UTC)

内存至少3G，处理器2核

3.3 启动docker cli界面

首先双击Kitematic (Alpha)，选择VirtualBox进入docker，之后进入

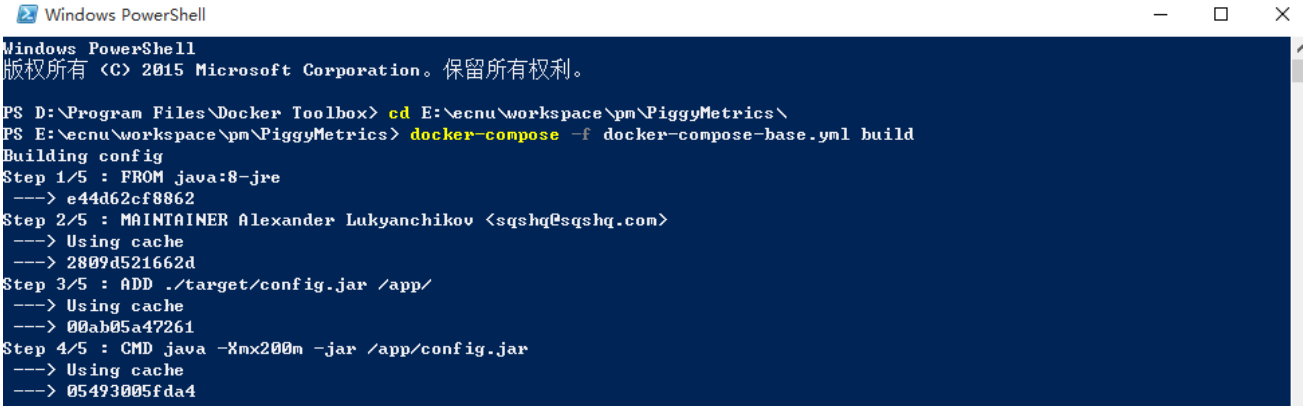


```
com.netflix.discovery.DiscoveryClient    :
DiscoveryClient_GATEWAY/775f42b93ac5:gateway:4000 - Re-
registering apps/GATEWAY
2017-06-27 04:56:55.856  INFO 1 --- [tbeatExecutor-0]
com.netflix.discovery.DiscoveryClient    :
DiscoveryClient_GATEWAY/775f42b93ac5:gateway:4000: registering
service...
2017-06-27 04:56:55.857  INFO 1 --- [freshExecutor-0]
com.netflix.discovery.DiscoveryClient    : The response status
is 200
2017-06-27 04:56:55.932  INFO 1 --- [tbeatExecutor-0]
com.netflix.discovery.DiscoveryClient    :
DiscoveryClient_GATEWAY/775f42b93ac5:gateway:4000 - registration
status: 204
2017-06-27 04:56:55.936  INFO 1 --- [nfoReplicator-0]
com.netflix.discovery.DiscoveryClient    :
DiscoveryClient_GATEWAY/775f42b93ac5:gateway:4000 - registration
status: 204
2017-06-27 04:57:25.880  INFO 1 --- [freshExecutor-0]
com.netflix.discovery.DiscoveryClient    : Disable delta
property : false
2017-06-27 04:57:25.881  INFO 1 --- [freshExecutor-0]
com.netflix.discovery.DiscoveryClient    : Single vip registry
refresh property : null
2017-06-27 04:57:25.881  INFO 1 --- [freshExecutor-0]
com.netflix.discovery.DiscoveryClient    : Force full registry
fetch : false
2017-06-27 04:57:25.881  INFO 1 --- [freshExecutor-0]
com.netflix.discovery.DiscoveryClient    : Application is null :
false
2017-06-27 04:57:25.881  INFO 1 --- [freshExecutor-0]
```

You can access this container using the following IP address and port:

DOCKER PORT	ACCESS URL
4000/tcp	192.168.99.100:80

点击左下角的docker cli



```
Windows PowerShell
版权所有 (C) 2015 Microsoft Corporation。保留所有权利。

PS D:\Program Files\Docker Toolbox> cd E:\ecnu\workspace\pm\PiggyMetrics\
PS E:\ecnu\workspace\pm\PiggyMetrics> docker-compose -f docker-compose-base.yml build
Building config
Step 1/5 : FROM java:8-jre
 ---> e44d62cf8862
Step 2/5 : MAINTAINER Alexander Lukyanchikov <sqshq@sqshq.com>
 ---> Using cache
 ---> 2809d521662d
Step 3/5 : ADD ./target/config.jar /app/
 ---> Using cache
 ---> 00ab05a47261
Step 4/5 : CMD java -Xmx200m -jar /app/config.jar
 ---> Using cache
 ---> 05493005fda4
```

下面可以开始编译运行微服务了

4、微服务编译运行

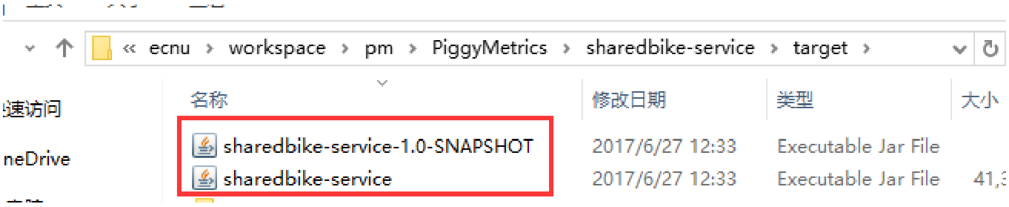
4.1 首先切换到源码目录 ，我的是cd E:\ecnu\workspace\pm\PiggyMetrics\

源码是maven构建

4.2 打包

mvn package -DskipTests

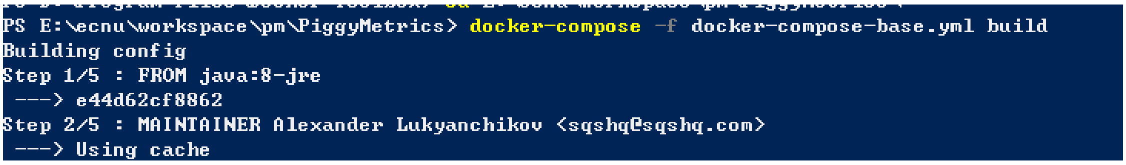
执行后在每一个项目的target下有两个jar包


« ecnu › workspace › pm › PiggyMetrics › sharedbike-service › target ›
速访问
neDrive

名称	修改日期	类型	大小
sharedbike-service-1.0-SNAPSHOT	2017/6/27 12:33	Executable Jar File	
sharedbike-service	2017/6/27 12:33	Executable Jar File	41,3

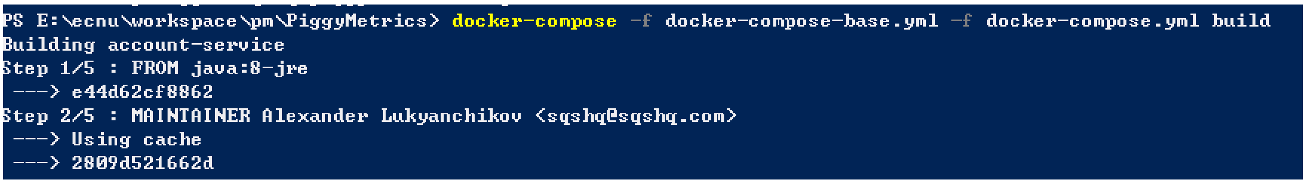
4.3 编译

docker-compose -f docker-compose-base.yml build



```
PS E:\ecnu\workspace\pm\PiggyMetrics> docker-compose -f docker-compose-base.yml build
Building config
Step 1/5 : FROM java:8-jre
 ---> e44d62cf8862
Step 2/5 : MAINTAINER Alexander Lukyanchikov <sqshq@sqshq.com>
 ---> Using cache
```

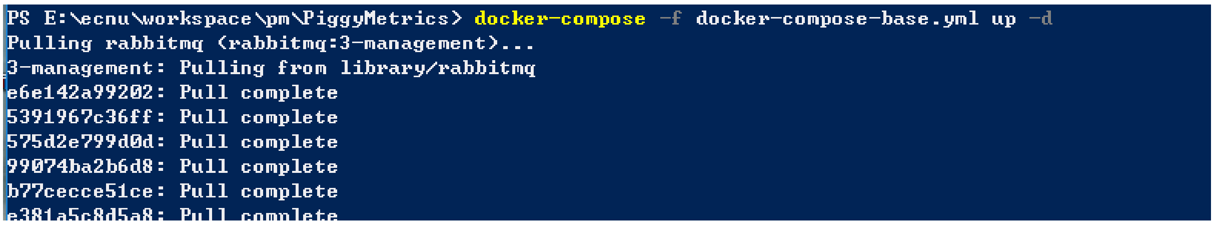
docker-compose -f docker-compose-base.yml -f docker-compose.yml build



```
PS E:\ecnu\workspace\pm\PiggyMetrics> docker-compose -f docker-compose-base.yml -f docker-compose.yml build
Building account-service
Step 1/5 : FROM java:8-jre
 ---> e44d62cf8862
Step 2/5 : MAINTAINER Alexander Lukyanchikov <sqshq@sqshq.com>
 ---> Using cache
 ---> 2809d521662d
```

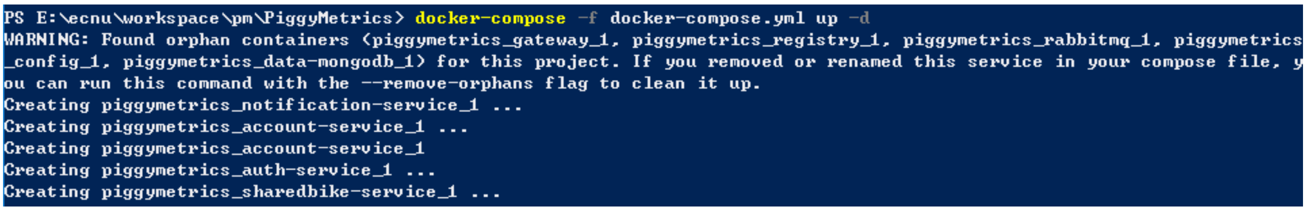
4.4 运行

docker-compose -f docker-compose-base.yml up -d



```
PS E:\ecnu\workspace\pm\PiggyMetrics> docker-compose -f docker-compose-base.yml up -d
Pulling rabbitmq (rabbitmq:3-management)...
3-management: Pulling from library/rabbitmq
e6e142a99202: Pull complete
5391967c36ff: Pull complete
575d2e799d0d: Pull complete
99074ba2b6d8: Pull complete
b77cecce51ce: Pull complete
e381a5c8d5a8: Pull complete
```

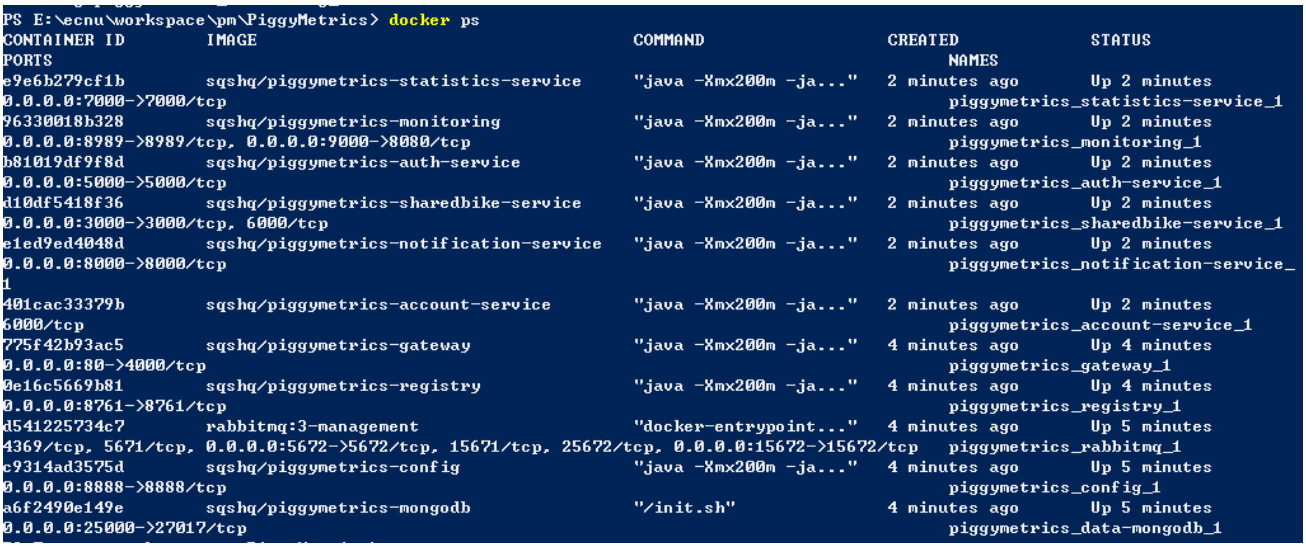
docker-compose -f docker-compose.yml up -d



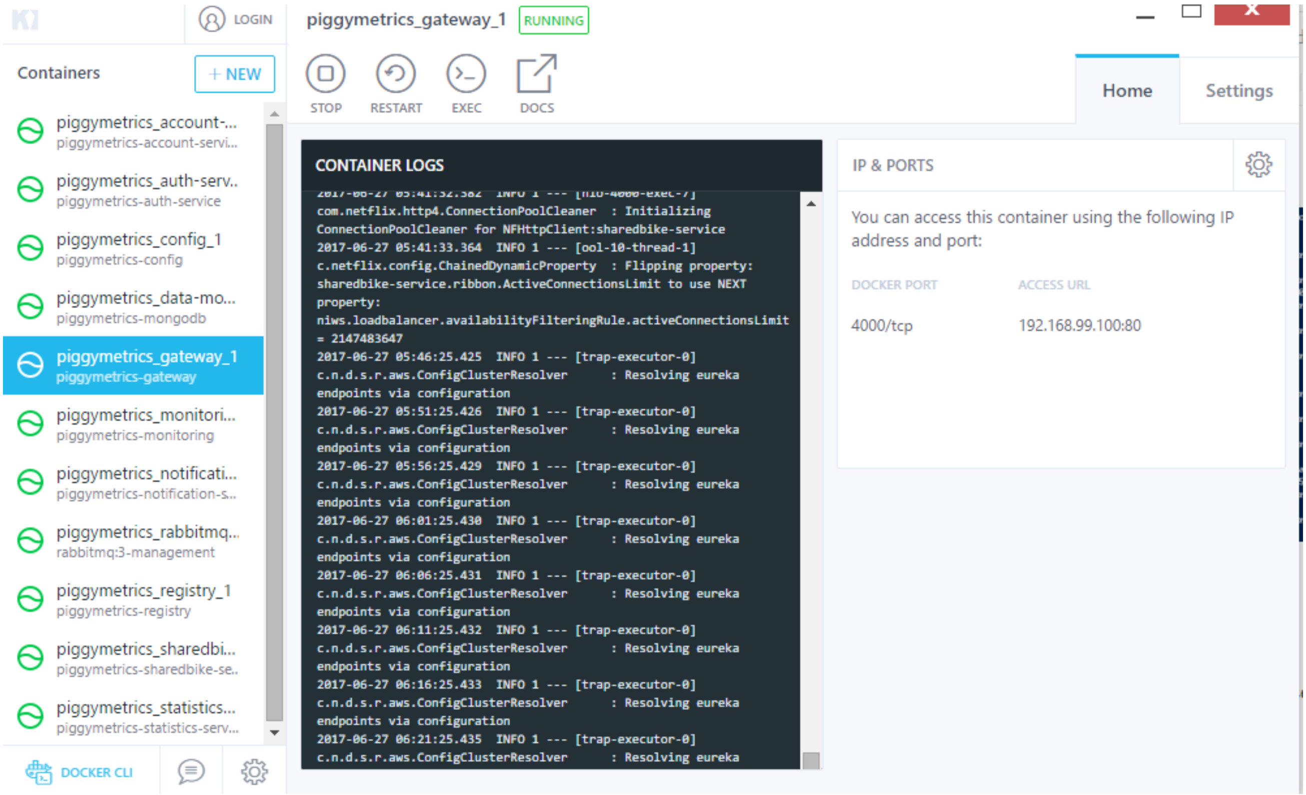
```
PS E:\ecnu\workspace\pm\PiggyMetrics> docker-compose -f docker-compose.yml up -d
WARNING: Found orphan containers (piggymetrics_gateway_1, piggymetrics_registry_1, piggymetrics_rabbitmq_1, piggymetrics
_config_1, piggymetrics_data-mongodb_1) for this project. If you removed or renamed this service in your compose file, y
ou can run this command with the --remove-orphans flag to clean it up.
Creating piggymetrics_notification-service_1 ...
Creating piggymetrics_account-service_1 ...
Creating piggymetrics_account-service_1
Creating piggymetrics_auth-service_1 ...
Creating piggymetrics_sharedbike-service_1 ...
```

查看启动的容器

docker ps



```
PS E:\ecnu\workspace\pm\PiggyMetrics> docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED             STATUS
PORTS                                                                                        NAMES
e9e6b279cf1b        sqshq/piggymetrics-statistics-service     "java -Xmx200m -ja..."   2 minutes ago       Up 2 minutes
0.0.0.0:7000->7000/tcp                                                                       piggymetrics_statistics-service_1
96330018b328        sqshq/piggymetrics-monitoring             "java -Xmx200m -ja..."   2 minutes ago       Up 2 minutes
0.0.0.0:8989->8989/tcp, 0.0.0.0:9000->8080/tcp                                               piggymetrics_monitoring_1
b81019df9f8d        sqshq/piggymetrics-auth-service           "java -Xmx200m -ja..."   2 minutes ago       Up 2 minutes
0.0.0.0:5000->5000/tcp                                                                       piggymetrics_auth-service_1
d10df5418f36        sqshq/piggymetrics-sharedbike-service     "java -Xmx200m -ja..."   2 minutes ago       Up 2 minutes
0.0.0.0:3000->3000/tcp, 6000/tcp                                                             piggymetrics_sharedbike-service_1
e1ed9ed4048d        sqshq/piggymetrics-notification-service   "java -Xmx200m -ja..."   2 minutes ago       Up 2 minutes
0.0.0.0:8000->8000/tcp                                                                       piggymetrics_notification-service_
1
401cac33379b        sqshq/piggymetrics-account-service        "java -Xmx200m -ja..."   2 minutes ago       Up 2 minutes
6000/tcp                                                                                     piggymetrics_account-service_1
775f42b93ac5        sqshq/piggymetrics-gateway                "java -Xmx200m -ja..."   4 minutes ago       Up 4 minutes
0.0.0.0:80->4000/tcp                                                                         piggymetrics_gateway_1
0e16c5669b81        sqshq/piggymetrics-registry               "java -Xmx200m -ja..."   4 minutes ago       Up 4 minutes
0.0.0.0:8761->8761/tcp                                                                       piggymetrics_registry_1
d541225734c7        rabbitmq:3-management                     "docker-entrypoint..."   4 minutes ago       Up 5 minutes
4369/tcp, 5671/tcp, 0.0.0.0:5672->5672/tcp, 15671/tcp, 25672/tcp, 0.0.0.0:15672->15672/tcp   piggymetrics_rabbitmq_1
c9314ad3575d        sqshq/piggymetrics-config                 "java -Xmx200m -ja..."   4 minutes ago       Up 5 minutes
0.0.0.0:8888->8888/tcp                                                                       piggymetrics_config_1
a6f2490e149e        sqshq/piggymetrics-mongodb                "/init.sh"               4 minutes ago       Up 5 minutes
0.0.0.0:25000->27017/tcp                                                                     piggymetrics_data-mongodb_1
```



CONTAINER LOGS

```
2017-06-27 05:41:32.382  INFO 1 --- [nio-4000-exec-7]
com.netflix.http4.ConnectionPoolCleaner  : Initializing
ConnectionPoolCleaner for NFHttpClient:sharedbike-service
2017-06-27 05:41:33.364  INFO 1 --- [ool-10-thread-1]
c.netflix.config.ChainedDynamicProperty  : Flipping property:
sharedbike-service.ribbon.ActiveConnectionsLimit to use NEXT
property:
niws.loadbalancer.availabilityFilteringRule.activeConnectionsLimit
= 2147483647
2017-06-27 05:46:25.425  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 05:51:25.426  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 05:56:25.429  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:01:25.430  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:06:25.431  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:11:25.432  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:16:25.433  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:21:25.435  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
```

IP & PORTS

You can access this container using the following IP address and port:

DOCKER PORT	ACCESS URL
4000/tcp	192.168.99.100:80

服务器启动过程中，可能有些服务会启动失败

访问网站：http://192.168.99.100:8761/

最终会起以下6个服务，都必须全起，这几个网站才能全部访问。

http://192.168.99.100:8888/gateway/default　　user/admin

http://192.168.99.100:8888/registry/default

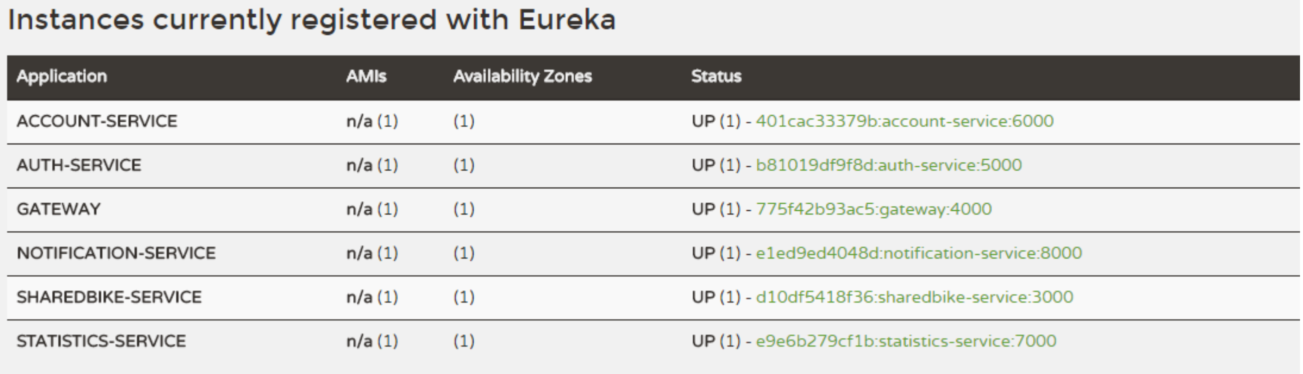
http://192.168.99.100:15672　　　　　guest/guest

http://192.168.99.100:8761

http://192.168.99.100

http://192.168.99.100:9000/hystrix

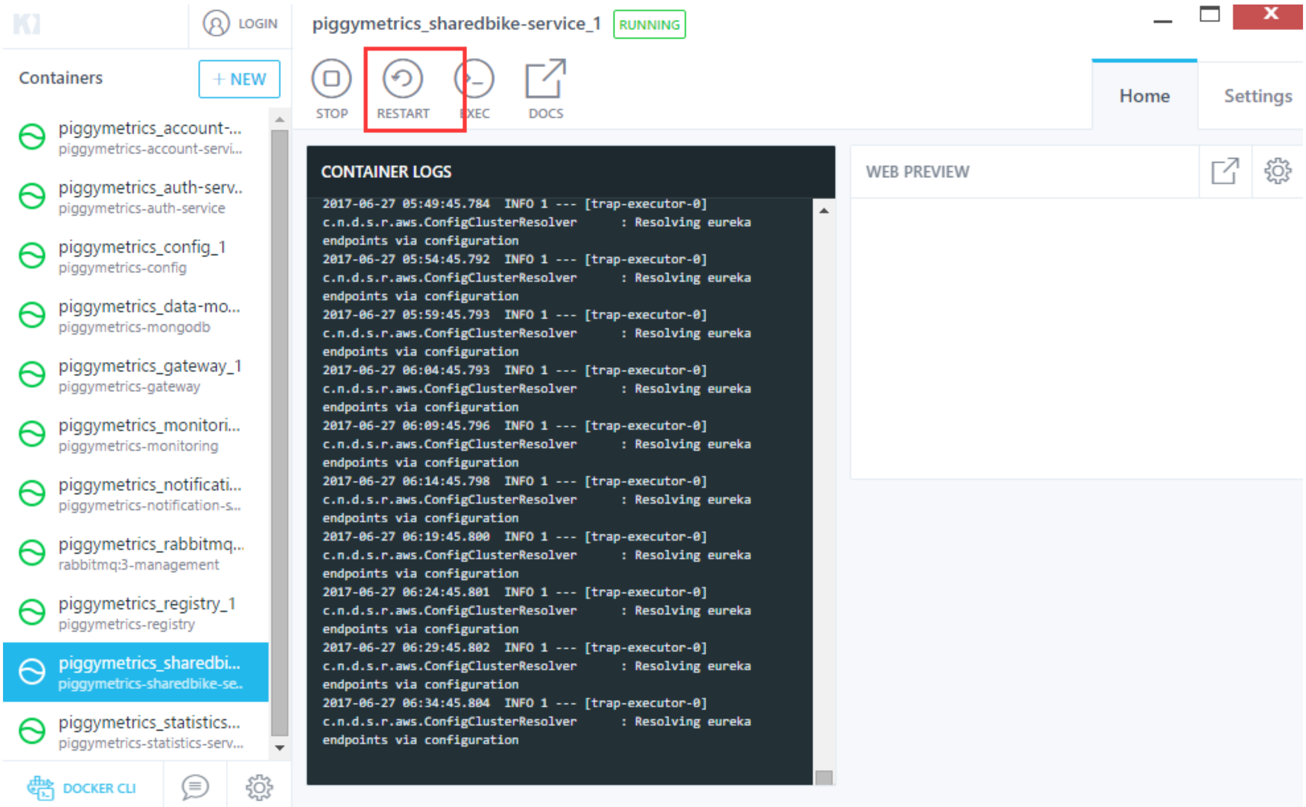
http://192.168.99.100:8989



Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
ACCOUNT-SERVICE	n/a (1)	(1)	UP (1) - 401cac33379b:account-service:6000
AUTH-SERVICE	n/a (1)	(1)	UP (1) - b81019df9f8d:auth-service:5000
GATEWAY	n/a (1)	(1)	UP (1) - 775f42b93ac5:gateway:4000
NOTIFICATION-SERVICE	n/a (1)	(1)	UP (1) - e1ed9ed4048d:notification-service:8000
SHAREDBIKE-SERVICE	n/a (1)	(1)	UP (1) - d10df5418f36:sharedbike-service:3000
STATISTICS-SERVICE	n/a (1)	(1)	UP (1) - e9e6b279cf1b:statistics-service:7000

如果发现有个服务没起，可以在restart，方法如图



```
2017-06-27 05:49:45.784  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 05:54:45.792  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 05:59:45.793  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:04:45.793  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:09:45.796  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:14:45.798  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:19:45.800  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:24:45.801  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:29:45.802  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
2017-06-27 06:34:45.804  INFO 1 --- [trap-executor-0]
c.n.d.s.r.aws.ConfigClusterResolver      : Resolving eureka
endpoints via configuration
```

5、接口访问

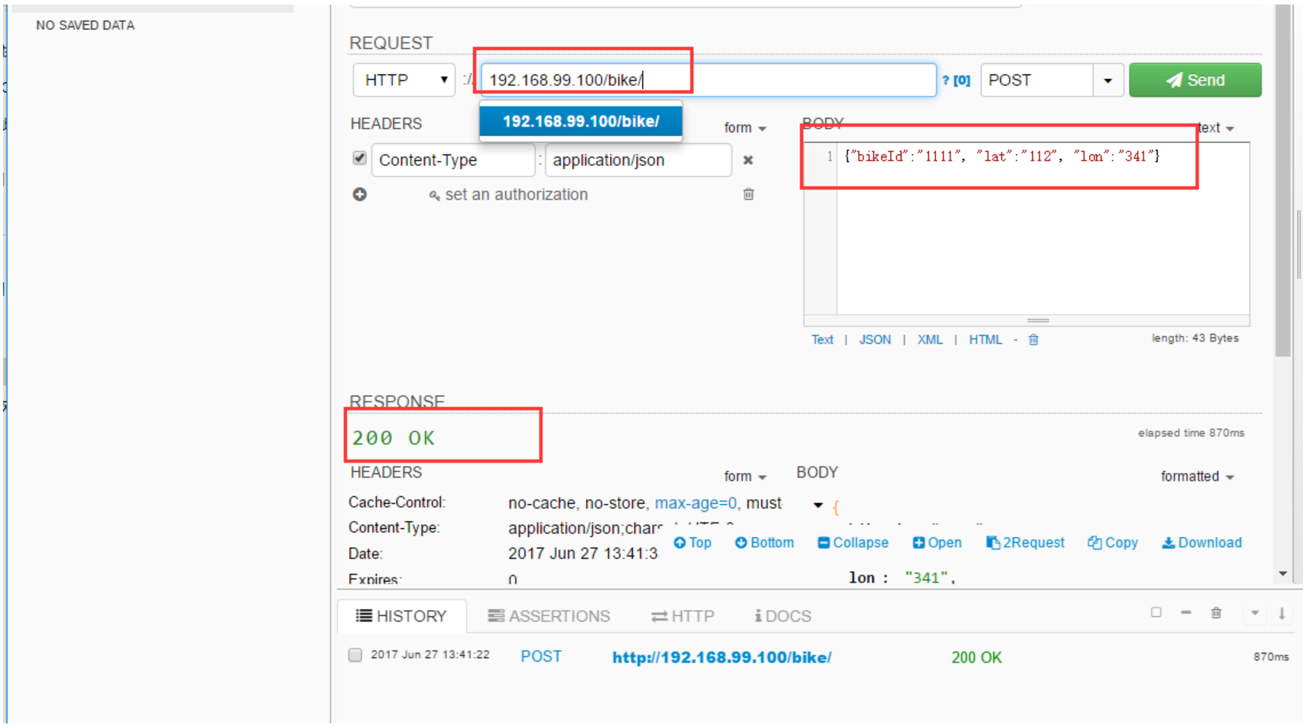
系统中新建了sharedbike-service项目，其中一个接口访问，如下图，首先得安装DHC插件

URL：192.168.99.100/bike/

POST方式

请求参数JSON格式：{"bikeId":"1111", "lat":"112", "lon":"341"}

200：代表请求成功



URL: 192.168.99.100/bike/ — POST

Content-Type : application/json

```
{"bikeId":"1111", "lat":"112", "lon":"341"}
```

length: 43 Bytes

200 OK

elapsed time 870ms

Cache-Control: no-cache, no-store, max-age=0, must
Content-Type: application/json;char
Date: 2017 Jun 27 13:41:3
Expires: 0

lon : "341",

2017 Jun 27 13:41:22	POST	http://192.168.99.100/bike/	200 OK	870ms

6、访问mongodb

6.1 获取mongodb帐号密码

使用工具robomongo客户端连接mongodb

默认的用户名密码可以查看mongodb目录下的init.sh

url：192.168.99.100

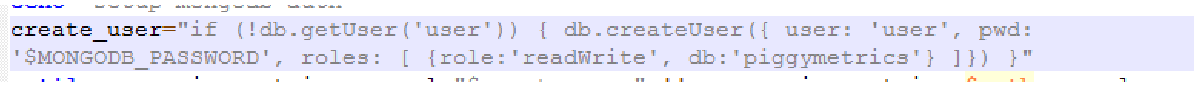
端口：25000

用户名：user

密码：admin

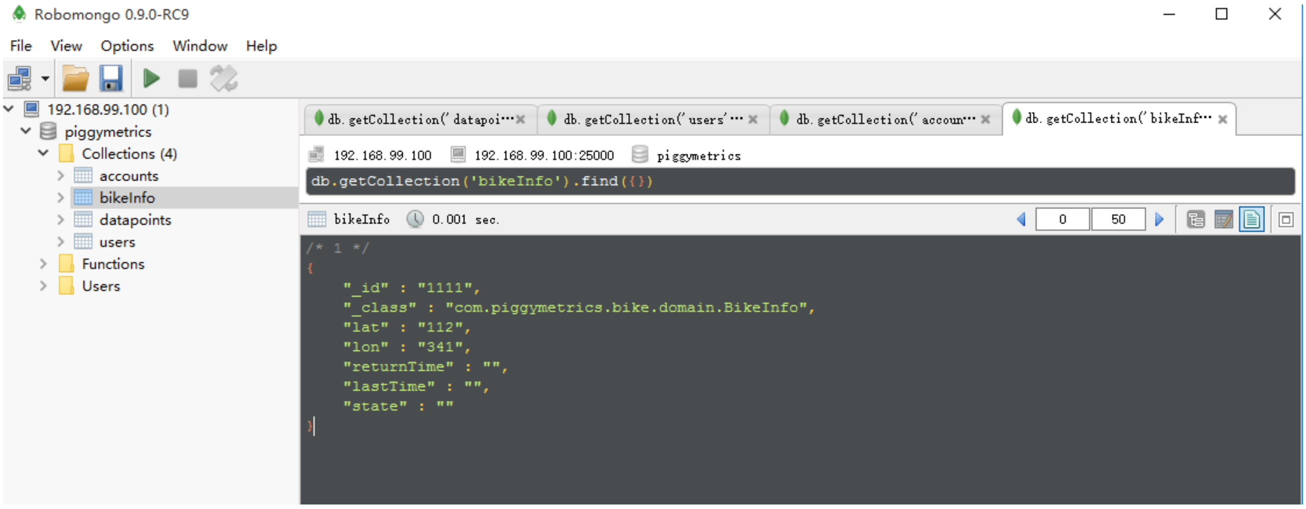
数据库名：piggymetrics

即如下



```
create_user="if (!db.getUser('user')) { db.createUser({ user: 'user', pwd:
'$MONGODB_PASSWORD', roles: [ {role:'readWrite', db:'piggymetrics'} ]}) }"
```

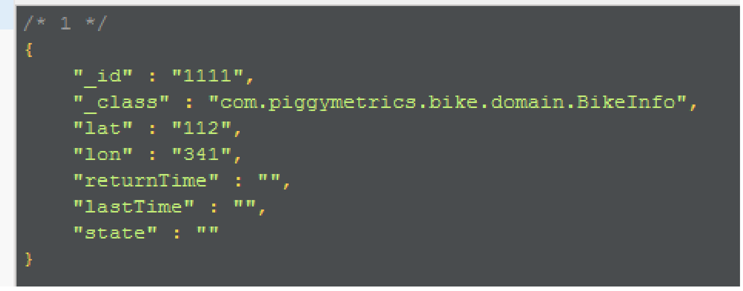
6.2 连接成功后：



```
db.getCollection('bikeInfo').find({})
```

```
/* 1 */
{
    "_id" : "1111",
    "_class" : "com.piggymetrics.bike.domain.BikeInfo",
    "lat" : "112",
    "lon" : "341",
    "returnTime" : "",
    "lastTime" : "",
    "state" : ""
}
```

插入的bikeinfo



```
/* 1 */
{
    "_id" : "1111",
    "_class" : "com.piggymetrics.bike.domain.BikeInfo",
    "lat" : "112",
    "lon" : "341",
    "returnTime" : "",
    "lastTime" : "",
    "state" : ""
}
```

参考网站：

http://git.oschina.net/huangmeng4520/PiggyMetrics/

http://www.jianshu.com/p/e26570d1b297?mType=Group