# Monasca安装配置指导

**作者：**张航东

Monasca作为Openstack的monitoring-as-a-service组件，目前社区和网上的资料还是比较少。本文通过作者团队的实践，记录了Monasca相关的安装和配置方法，供大家参考学习。

总的来说，Monasca通常需要以下几个部件提供服务：

* python-monascaclient (client)
* monasca-common (通用类)
* monasca-api (RESTful API)
* monasca-agent (从host上抓取计量数据发送给monasca)
* monasca-persister (接收、处理监控告警数据，并保存后端存储)
* monasca-thresh (计算阈值，发布告警)
* monasca-ui (UI界面)
* monasca-notification (接收告警，通知用户，如邮件) (本文未涉及)
* monasca-ceilometer (将ceilometer统计到的信息发送给monasca-api) (本文未涉及)

### 依赖组件

##### JAVA

**下载（jdk-8u144-linux-x64.rpm）：**<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

**安装：**

#yum install jdk-8u144-linux-x64.rpm

**配置环境变量 /etc/profile， 中添加如下内容：**

#set java environment

JAVA\_HOME=/usr/java/jdk1.8.0\_144

JRE\_HOME=${JAVA\_HOME}/jre

PATH=${PATH}:${JAVA\_HOME}/bin

CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

export JAVA\_HOME JRE\_HOME CLASSPATH PATH

**使配置生效：**

#Source /etc/profile

##### zookeeper

**下载：**<http://mirror.bit.edu.cn/apache/zookeeper/stable/zookeeper-3.4.10.tar.gz>

**解压：**

#tar -zxvf zookeeper-3.4.10.tar.gz

#cd zookeeper-3.4.10

**配置：**

#mv conf/zoo\_sample.cfg conf/zoo.cfg

#vi conf/zoo.cfg

**添加如下内容：**

tickTime=2000

dataDir=/var/zookeeper

clientPort=2181

**启动：**

#bin/zkServer.sh start

##### kafka

（参考官网指导：<http://kafka.apache.org/quickstart>）

**下载：**<http://mirror.bit.edu.cn/apache/kafka/0.11.0.0/kafka_2.12-0.11.0.0.tgz>

**解压：**

#tar -xzf kafka\_2.12-0.11.0.0.tgz

#cd kafka\_2.12-0.11.0.0

**启动Kafka服务：**

#bin/kafka-server-start.sh config/server.properties

**创建monasca-api需要的topic：**

#bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 64 --topic metrics

#bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 12 --topic events

#bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 12 --topic alarm-state-transitions

#bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 12 --topic alarm-notifications

#bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 8 --topic retry-notifications

#bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 8 --topic 60-seconds-notifications

##### Influxdb

（参考官网指导：<https://portal.influxdata.com/downloads>）

**下载：**<https://dl.influxdata.com/influxdb/releases/influxdb-1.3.1.x86_64.rpm>

**安装：**

#yum install influxdb-1.3.1.x86\_64.rpm

**启动：**

#systemctl start influxdb

**进入influxdb：**

#influx

**创建monasca数据库：**

>CREATE DATABASE monasca

**设置数据保留时间：**

>CREATE RETENTION POLICY " persister\_all" ON "monasca" DURATION 90d REPLICATION 1

**创建Monasca用户：**

>USE monasca

>CREATE USER "monasca" WITH PASSWORD 'monasca'WITH ALL PRIVILEGES

**同时，安装influxdb python环境：**

#pip install influxdb

### python-monascaclient

#pip install python-monascaclient

### monasca-common

#git clone https://github.com/openstack/monasca-common.git

#cd monasca-common

#python setup.py install

### monasca-api

##### 安装

#git clone https://github.com/openstack/monasca-api.git

#cd monasca-api

#python setup.py install

##### 配置

* + 1. Keystone相关配置

1. **创建monasca用户**

#openstack user create --domain default --password-prompt monasca

#openstack role add --project service --user monasca admin

1. **创建monasca服务**

#openstack service create --name monasca --description "This is monasca service" monitoring

1. **创建endpoint**

#openstack endpoint create --region RegionOne monitoring public http://10.127.2.121:8070/v2.0

#openstack endpoint create --region RegionOne monitoring internal http://10.127.2.121:8070/v2.0

#openstack endpoint create --region RegionOne monitoring admin http://10.127.2.121:8070/v2.0

* + 1. Wsgi相关配置

在/etc/httpd/conf.d/目录下，创建wsgi-monasca.conf文件，并添加如下内容：

Listen 10.127.2.121:8070

<VirtualHost \*:8070>

WSGIDaemonProcess monasca\_api processes=2 threads=4 user=monasca display-name=%{GROUP}

WSGIProcessGroup monasca\_api

WSGIScriptAlias / /usr/lib/python2.7/site-packages/monasca\_api/api/wsgi.py

WSGIApplicationGroup %{GLOBAL}

<Directory /usr/lib/python2.7/site-packages/monasca\_api>

Require all granted

</Directory>

ErrorLog /var/log/monasca/api/wsgi.log

CustomLog /var/log/monasca/api/wsgi-access.log combined

</VirtualHost>

WSGISocketPrefix /var/run/httpd

* + 1. Monasca配置

**创建Monasca日志目录**

#mkdir /var/log/monasca

#mkdir /var/log/monasca/api

#touch /var/log/monasca/api/monasca-api.log

在 /etc/monasca/api-config.conf文件内添加如下内容：

[DEFAULT]

region = RegionOne

[security]

default\_authorized\_roles = admin

agent\_authorized\_roles = admin, user

read\_only\_authorized\_roles =admin, user

delegate\_authorized\_roles = admin

[kafka]

uri = 127.0.0.1

[influxdb]

ip\_address = 127.0.0.1

port = 8086

user = monasca

password = monasca

database\_name = monasca

[database]

connection = mysql+pymysql://monasca:monasca@10.127.2.121/monasca?charset=utf8mb4

[keystone\_authtoken]

auth\_uri = http://10.127.2.136:5000

auth\_url = http://10.127.2.136:35357

auth\_type = password

project\_domain\_name = default

user\_domain\_name = default

project\_name = service

username = monasca

password = monasca

insecure=false

* + 1. Mysql 数据库配置

**进入Mysql：**

#mysql -u root -p

**执行以下脚本：**

# mysql -u root -p < path/to/sql/mon\_mysql.sql



### monasca-agent

（参考：<https://github.com/openstack/monasca-agent/blob/master/docs/Agent.md>）

##### 安装

#pip install monasca-agent

##### 启动

#monasca-setup **--username** admin **--password** 123456 **--project\_name** admin **--project\_domain\_id** 9f95b9967b894c928880feb32fad1d0d **--user\_domain\_id** 9f95b9967b894c928880feb32fad1d0d **--keystone\_url** http://10.127.2.136:35357/v3 --system\_only

PS：**黑色粗体**为keystone鉴权需要的参数，根据实际情况替换后面红色的value。另外，在启动后，可能会出现服务启动失败的情况，日志中会显示问题（大多是依赖包版本问题），根据情况修正即可。

### monasca-persister

##### 安装

#git clone https://github.com/openstack/monasca-persister.git

#cd monasca-persister

#python setup.py install

##### 配置

**创建persister日志目录**

#mkdir /var/log/monasca/persister

#touch /var/log/monasca/persister/persister.log

**修改配置文件 /etc/monasca/persister.conf**，内添加如下内容：

[zookeeper]

uri = 127.0.0.1:2181

[kafka\_alarm\_history]

uri = 127.0.0.1:9092

[kafka\_metrics]

uri = 127.0.0.1:9092

[influxdb]

database\_name = monasca

ip\_address = 127.0.0.1

port = 8086

user = monasca

password = monasca

##### 启动

#monasca-persister

### monasca-thresh

PS：monasca-thresh是JAVA工程，所以需要maven工具进行先进行编译

##### 安装 apache-maven

**下载（apache-maven-3.5.0-bin.tar.gz）：**<http://maven.apache.org/download.cgi>

**解压：**

#tar -xvf apache-maven-3.5.0-bin.tar.gz

目录内的bin/mvn就是编译monasca-thresh工程需要的工具，

直接使用，或添加到环境变量，修改 /etc/profile中，添加如下内容：

MAVEN\_HOME=/usr/local/apache-maven-3.5.0

export MAVEN\_HOME

export PATH=${PATH}:${MAVEN\_HOME}/bin

**使配置生效：**

#source /etc/profile

##### 编译monasca-thresh工程

PS：monasca-thresh需要依赖monasca-common（基本所有monasca的组件都依赖common），common同时支持java和python，但是thresh目前只有java版本，所以我们需要先将common编译thresh可以引用的java包。

* + 1. 编译monasca-common

**进入monasca-common目录**

**给文件添加执行权限：**

#chmod +x run\_maven.sh

**修改文件格式（dos改为unix）：**

#vi run\_maven.sh

… …

:set ff=unix

**（保存退出）**

**执行mvn命令编译：**

#mvn clean install

* + 1. 编译monasca-thresh

**下载：**

#git clone <https://github.com/openstack/monasca-thresh.git>

#cd monasca-thresh

**给文件添加执行权限：**

#chmod +x run\_maven.sh

**修改文件格式（dos改为unix）：**

#vi run\_maven.sh

… …

:set ff=unix

**（保存退出）**

**进入thresh目录：**

#cd thresh

**执行mvn命令编译：**

#mvn package

执行完后， ./target/monasca-thresh-x.x.x-SNAPSHOT-shaded.jar 便是我们需要的thresh运行程序。

不过，这个程序并不能直接运行，需要运行在apache storm平台上。

##### 安装apache storm

**下载（apache-storm-1.1.1.tar.gz）：**<http://storm.apache.org/downloads.html>

**解压：**

#tar xvf apache-storm-1.1.1.tar.gz

#cd apache-storm-1.1.1

**修改配置文件 /conf/storm.yaml**，添加如下内容：

########### These MUST be filled in for a storm configuration

storm.zookeeper.servers:

- "127.0.0.1"

nimbus.seeds: ["127.0.0.1"]

**启动：**

#bin/storm nimbus

#bin/storm supervisor

##### 启动monasca-thresh

在7.2中，我们已经成功编译除了monasca-thresh运行的jar包（…/monasca-thresh/thresh/target/ monasca-thresh-x.x.x-SNAPSHOT-shaded.jar）

这一步，我们要基于apache storm平台，来启动monasca-thresh。

* + 1. 配置

**进去monasca-thresh工程目录**

**拷贝配置文件模板到 etc/monasca/目录下**

#cp thresh/src/main/config/thresh-sample-config.yml /etc/monasca/thresh.conf

**修改配置文件/etc/monasca/thresh.conf**，修改一下几项**：**

…

zookeeperConnect: 127.0.0.1:2181 (多处)

…

metadataBrokerList: 127.0.0.1:9092

…

database:

driverClass: "com.mysql.jdbc.jdbc2.optional.MysqlDataSource"

url: "jdbc:mysql://10.127.2.121/monasca?useSSL=true"

user: "monasca"

password: "monasca"

…

statsdConfig:  
 host: 127.0.0.1  
 port: 8125

* + 1. 启动

#.../apache-storm-1.1.1/bin/storm jar …/monasca-thresh/thresh/target/ monasca-thresh-x.x.x-SNAPSHOT-shaded.jar monasca.thresh.ThresholdingEngine /etc/monasca/thresh.conf thresh-cluster

PS：根据实际情况修改上述红色部分

### monasca-ui

##### 安装 horizon (源码方式)

参考：<https://docs.openstack.org/horizon/latest/install/from-source.html>

**下载安装**

#git clone https://github.com/openstack/horizon.git

#cd horizon

#pip install -r requirements.txt

PS：由于权限问题，尽量避免下载到/root目录，建议下载到/opt目录

**生成local\_settings.py**

#cp openstack\_dashboard/local/local\_settings.py.example openstack\_dashboard/local/local\_settings.py

**修改local\_settings.py，配置内容如下：**

COMPRESS\_OFFLINE = True

ALLOWED\_HOSTS = ['\*', ]

OPENSTACK\_API\_VERSIONS = {

"identity": 3,

"image": 2,

"volume": 2,

}

OPENSTACK\_KEYSTONE\_MULTIDOMAIN\_SUPPORT = True

OPENSTACK\_KEYSTONE\_DEFAULT\_DOMAIN = 'Default'

OPENSTACK\_HOST = "10.127.2.136"

OPENSTACK\_KEYSTONE\_URL = "http://%s:5000/v3" % OPENSTACK\_HOST

OPENSTACK\_KEYSTONE\_DEFAULT\_ROLE = "\_member\_"

**安装语言包**

#apt-get install gettext

#./manage.py compilemessages

**静态文件压缩**

#./manage.py collectstatic

#./manage.py compress

**生成wsgi配置文件horizon.conf**

#./manage.py make\_web\_conf --apache > /etc/httpd/conf.d/horizon.conf

**修改horizon.conf，配置内容如下：**

<VirtualHost \*:80>

ServerAdmin webmaster@openstack.org

ServerName openstack\_dashboard

DocumentRoot /opt/horizon/

LogLevel warn

ErrorLog /var/log/httpd/openstack\_dashboard-error.log

CustomLog /var/log/httpd/openstack\_dashboard-access.log combined

WSGIScriptReloading On

WSGIDaemonProcess openstack\_dashboard\_website

WSGIProcessGroup openstack\_dashboard\_website

WSGIApplicationGroup openstack\_dashboard\_website

WSGIPassAuthorization On

WSGIScriptAlias / /opt/horizon/openstack\_dashboard/wsgi/horizon.wsgi

<Location "/">

Require all granted

</Location>

Alias /static /opt/horizon/static

<Location "/static">

SetHandler None

</Location>

</Virtualhost>

**重启httpd服务：**

#systemctl restart httpd

##### 安装 monasca-ui

参考：<https://github.com/openstack/monasca-ui>

**在horizon的requirements.txt中添加对monasca-ui的支持**

进入已下载的horizon工程的目录，

#vi requirements.txt

在末尾添加如下内容：

… …

git+https://git.openstack.org/openstack/monasca-ui.git

再次执行命令（安装horzion时执行过一次）：

#pip install -r requirements.txt

**将monasca链接到horizon**

#ln -sf $(pwd)/../monasca-ui/monitoring/enabled/\_50\_admin\_add\_monitoring\_panel.py $(pwd)/openstack\_dashboard/enabled/\_50\_admin\_add\_monitoring\_panel.py

#ln -sf $(pwd)/../monasca-ui/monitoring/conf/monitoring\_policy.json $(pwd)/openstack\_dashboard/conf/monitoring\_policy.json

#ln -sfF $(pwd)/../monasca-ui/monitoring $(pwd)/monitoring

**重启httpd服务：**

#systemctl restart httpd