Ansible Installation for Vagrant modifyied !

| Observation(Error) | Hypothesis(Remark) | Experimentation(Action) |
| --- | --- | --- |
| NOTIFIED: [kafka | restart kafka] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  failed: [mini-mon] => {"failed": true}  msg: stop: Unknown instance:  start: Job failed to start |  | sudo /opt/kafka/bin/kafka-server-start.sh /etc/kafka/server.properties |
| java.lang.IllegalArgumentException: requirement failed: Missing required property 'broker.id' |  | mini\_mon\_local was set to old ip address, changed to new one |
| java.lang.IllegalArgumentException: requirement failed: Missing required property 'broker.id' |  | Changed local\_monasca\_ip to localhost |
| java.lang.IllegalArgumentException: requirement failed: Missing required property 'broker.id' |  | 127.0.0.1 monasca  127.0.0.1 mini-mon  in /etc/hosts |
| java.lang.IllegalArgumentException: requirement failed: Missing required property 'broker.id' |  | Changed mini\_mon\_local to IP |
| java.lang.IllegalArgumentException: requirement failed: Missing required property 'broker.id' | kafka\_hosts = 10.11.12.5  ansible\_fqdn =localhost | CHANGED: line 9 in host\_vars/mini\_mon : {{mini\_mon\_local}} --> localhost |
| failed: [mini-mon] => {"failed": true}  msg: stop: Unknown instance:  start: Job failed to start  ????? | WARN Property C is not valid (kafka.utils.VerifiableProperties)  WARN Property E is not valid (kafka.utils.VerifiableProperties) | Delete: --limit @/home/jaafar/mini-mon.retry  delete E,C comment |
| failed: [mini-mon] => {"failed": true}  msg: stop: Unknown instance:  start: Job failed to start  ????? |  | sudo apt-get install default-jre  changed zooker to localhost  and add 9092 to OS secur. list |
| failed: [mini-mon] => {"failed": true}  msg: stop: Unknown instance:  start: Job failed to start  ????? |  |  |
| SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".  SLF4J: Defaulting to no-operation (NOP) logger implementation  SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details. |  | There is a documentation in [SLf4J site](http://www.slf4j.org/codes.html" \l "StaticLoggerBinder) which resolves this. I [ubuntu@monasca](mailto:ubuntu@monasca):/usr/share/maven-repo/org/slf4j/slf4j-api/1.7.5  $ ls  CatalogJar.htm slf4j-api-1.7.5.jar slf4j-api-1.7.5.pom slf4j-simple-1.7.5.jar |
| SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".  SLF4J: Defaulting to no-operation (NOP) logger implementation |  | Giving Up and rebuilind monasca.  Good night. |
| [https://yaohuangportal.wordpress.com/2014/06/24/kafka-console-producer-sh-kafka\_2-10-0-8-1-reported-slf4j-failed-to-load-class-org-slf4j-impl-staticloggerbinder/#comment-10](https://yaohuangportal.wordpress.com/2014/06/24/kafka-console-producer-sh-kafka_2-10-0-8-1-reported-slf4j-failed-to-load-class-org-slf4j-impl-staticloggerbinder/" \l "comment-10) | It worked |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Manual Installation: <https://www.shaunos.com/643.html>

Username: monasca

Password: qydcos → password

Command/ Installation Process: Start date : ### Feb 18, 16:13

install packages and tools we needed:

apt-get install -y git

git clone <https://github.com/shaunos/monasca.git>

sudo apt-get install openjdk-7-jre-headless python-pip python-dev

install mysql database:

apt-get install -y mysql-server

# Password for mySQL root is password

# create monasca database schema, download [mon\_mysql](https://www.shaunos.com/wp-content/uploads/2016/09/mon_mysql.zip) here, the schema file in #github has a bug, and it can not create notification, i have fixed it here. #remember to modify the user name and password in line 234,235 of mon\_mysql.sql #to yours.

# mysql\_schema link: <https://www.shaunos.com/wp-content/uploads/2016/09/mon_mysql.zip>

wget https://www.shaunos.com/wp-content/uploads/2016/09/mon\_mysql.zip –no-check-certificate

# You will find a copy of this file in the curret directory

sudo apt-get install -y unzip

unzip mon\_mysql.zip

# Modify the my\_sql password, changed line 234 and 235. password: password, #username: monasca

mysql -uroot -ppassword < mon\_mysql.sql

install zookeeper

sudo apt-get install -y zookeeper zookeeperd zookeeper-bin

sudo service zookeeper restart

install and configure kafka

wget <http://apache.mirrors.tds.net/kafka/0.8.1.1/kafka_2.9.2-0.8.1.1.tgz>

sudo mv kafka\_2.9.2-0.8.1.1.tgz /opt

cd /opt

sudo tar zxf kafka\_2.9.2-0.8.1.1.tgz

sudo ln -s /opt/kafka\_2.9.2-0.8.1.1/ /opt/kafka

sudo ln -s /opt/kafka/config /etc/kafka

# adding 127.0.0.1 monasca to /etc/hosts

# Create kafka system user

useradd kafka -U -r

#create kafka startup scripts in /etc/init/kafka.conf, copy following contents into /etc/init/kafka.conf #and save it.

*description "Kafka"*

*start on runlevel [2345]*

*stop on runlevel [!2345]*

*respawn*

*limit nofile 32768 32768*

*# If zookeeper is running on this box also give it time to start up properly*

*pre-start script*

*if [ -e /etc/init.d/zookeeper ]; then*

*/etc/init.d/zookeeper restart*

*fi*

*end script*

*# Rather than using setuid/setgid sudo is used because the pre-start task must run as root*

*exec sudo -Hu kafka -g kafka KAFKA\_HEAP\_OPTS="-Xmx1G -Xms1G" JMX\_PORT=9997 /opt/kafka/bin/kafka-server-start.sh /etc/kafka/server.properties*

**configure kafka, vim /etc/kafka/server.properties, make sure the following contents is configured**

host.name=localhost

advertised.host.name=localhost

log.dirs=/var/kafka

**create kafka log dirs.**

mkdir /var/kafka

mkdir /var/log/kafka

chown -R kafka. /var/kafka/

chown -R kafka. /var/log/kafka/

**sudo** service kafka start

the next step is to create kafka topics.

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

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

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

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

sudo /opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 12 --topic stream-definitions

sudo /opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 12 --topic transform-definitions

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

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

sudo /opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 12 --topic stream-notifications

sudo /opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 3 --topic retry-notifications

install and configure influxdb

sudo curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add -

echo "deb https://repos.influxdata.com/ubuntu trusty stable" | sudo tee /etc/apt/sources.list.d/influxdb.list

apt-get update

apt-get install -y apt-transport-https

apt-get install -y influxdb

service influxdb start

# create influxdb database, user, password, retention policy, change password to yours.

influx

CREATE DATABASE mon

CREATE USER monasca WITH PASSWORD 'qydcos'

CREATE RETENTION POLICY persister\_all ON mon DURATION 90d REPLICATION 1 DEFAULT

exit

install and configure storm

wget http://apache.mirrors.tds.net/storm/apache-storm-0.9.6/apache-storm-0.9.6.tar.gz

mkdir /opt/storm

cp apache-storm-0.9.6.tar.gz /opt/storm/

cd /opt/storm/

tar xzf apache-storm-0.9.6.tar.gz

ln -s /opt/storm/apache-storm-0.9.6 /opt/storm/current

sudo useradd storm -U -r

sudo mkdir /var/storm

sudo mkdir /var/log/storm

sudo chown -R storm. /var/storm/

sudo chown -R storm. /var/log/storm/

sudo mkdir -p storm/conf

modify storm.yaml as follow, vim current/storm/conf/storm.yaml

### base

java.library.path: "/usr/local/lib:/opt/local/lib:/usr/lib"

storm.local.dir: "/var/storm"

### zookeeper.\*

storm.zookeeper.servers:

- "localhost"

storm.zookeeper.port: 2181

storm.zookeeper.retry.interval: 5000

storm.zookeeper.retry.times: 29

storm.zookeeper.root: "/storm"

storm.zookeeper.session.timeout: 30000

### supervisor.\* configs are for node supervisors

supervisor.slots.ports:

- 6701

- 6702

- 6703

- 6704

supervisor.childopts: "-Xmx1024m"

### worker.\* configs are for task workers

worker.childopts: "-Xmx1280m -XX:+UseConcMarkSweepGC -Dcom.sun.management.jmxremote"

### nimbus.\* configs are for the masteri

nimbus.host: "localhost"

nimbus.thrift.port: 6627

mbus.childopts: "-Xmx1024m"

### ui.\* configs are for the master

ui.host: 127.0.0.1

ui.port: 8078

ui.childopts: "-Xmx768m"

### drpc.\* configs

### transactional.\* configs

transactional.zookeeper.servers:

- "localhost"

transactional.zookeeper.port: 2181

transactional.zookeeper.root: "/storm-transactional"

### topology.\* configs are for specific executing storms

topology.acker.executors: 1

topology.debug: false

logviewer.port: 8077

logviewer.childopts: "-Xmx128m"

create storm supervisor startup scripts, vim /etc/init/storm-supervisor.conf

# Startup script for Storm Supervisor

description "Storm Supervisor daemon"

start on runlevel [2345]

console log

respawn

kill timeout 240

respawn limit 25 5

setgid storm

setuid storm

chdir /opt/storm/current

exec /opt/storm/current/bin/storm supervisor

create storm nimbus scripts.vim /etc/init/storm-nimbus.conf

# Startup script for Storm Nimbus

description "Storm Nimbus daemon"

start on runlevel [2345]

console log

respawn

kill timeout 240

respawn limit 25 5

setgid storm

setuid storm

chdir /opt/storm/current

exec /opt/storm/current/bin/storm nimbus

# start storm supervisor and nimbus

sudo service storm-supervisor start

sudo service storm-nimbus start

install monasca api python packages  
some monasca components have both python and java code available, mainly i choose python code to deploy.

sudo pip install monasca-common

sudo pip install gunicorn

sudo pip install greenlet # Required for both

sudo pip install eventlet # For eventlet workers

sudo pip install gevent # For gevent workers

sudo pip install monasca-api

sudo pip install influxdb

sudo mkdir /etc/monasca

vim /etc/monasca/api-config.ini , modify host to your ip address

[DEFAULT]

name = monasca\_api

[pipeline:main]

# Add validator in the pipeline so the metrics messages can be validated.

pipeline = auth keystonecontext api

[app:api]

paste.app\_factory = monasca\_api.api.server:launch

[filter:auth]

paste.filter\_factory = keystonemiddleware.auth\_token:filter\_factory

[filter:keystonecontext]

paste.filter\_factory = monasca\_api.middleware.keystone\_context\_filter:filter\_factory

[server:main]

use = egg:gunicorn#main

host = 192.168.2.23

port = 8082

workers = 1

proc\_name = monasca\_api

vim /etc/monasca/api-config.conf, modify the following contents

[[DEFAULT]

# logging, make sure that the user under whom the server runs has permission

# to write to the directory.

log\_file = monasca-api.log

log\_dir = /var/log/monasca/api/

debug=False

region = RegionOne

[security]

# The roles that are allowed full access to the API.

default\_authorized\_roles = admin, user, domainuser, domainadmin, monasca-user

# The roles that are allowed to only POST metrics to the API. This role would be used by the Monasca Agen$

agent\_authorized\_roles = admin

# The roles that are allowed to only GET metrics from the API.

read\_only\_authorized\_roles = admin

# The roles that are allowed to access the API on behalf of another tenant.

# For example, a service can POST metrics to another tenant if they are a member of the "delegate" role.

delegate\_authorized\_roles = admin

[kafka]

# The endpoint to the kafka server

uri = localhost:9092

[influxdb]

# Only needed if Influxdb database is used for backend.

# The IP address of the InfluxDB service.

ip\_address = localhost

# The port number that the InfluxDB service is listening on.

port = 8086

# The username to authenticate with.

user = monasca

# The password to authenticate with.

password = password

# The name of the InfluxDB database to use.

database\_name = mon

[database]

url = "mysql+pymysql://monasca:password@127.0.0.1/mon"

[keystone\_authtoken]

identity\_uri = http://157.159.232.243:35357

auth\_uri = http://157.159.232.243:5000

admin\_password = password

admin\_user = admin

admin\_tenant\_name = admin

cafile =

certfile =

keyfile =

insecure = false

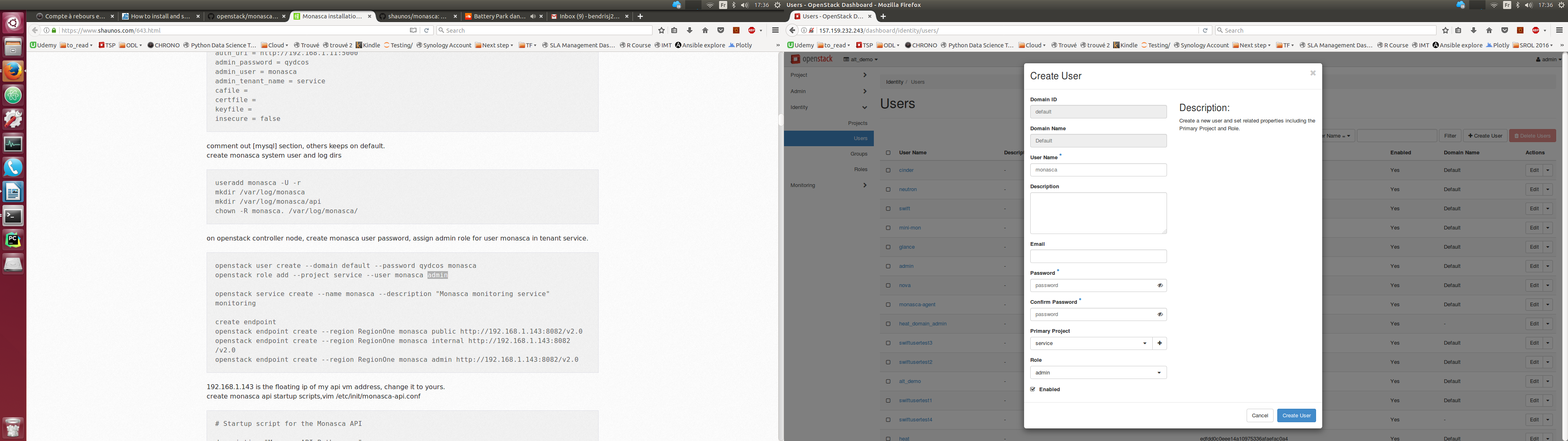
comment out [mysql] section, others keeps on default.  
create monasca system user and log dirs

sudo useradd monasca -U -r

sudo mkdir /var/log/monasca

sudo mkdir /var/log/monasca/api

sudo chown -R monasca. /var/log/monasca/



In devstack

[ubuntu@~(openstack)]$ cat keystonerc

unset OS\_SERVICE\_TOKEN

export OS\_USERNAME=admin #jaafar

export OS\_PASSWORD=password # marocfrance1

export OS\_AUTH\_URL=http://localhost:5000/v2.0 #http://157.159.232.52:5000/v2.0

export PS1='[\u@\W(openstack)]\$ '

export OS\_TENANT\_NAME=admin

export OS\_PROJECT\_NAME=admin

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

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

openstack service create --name monasca --description "Monasca monitoring service" monitoring

openstack endpoint create --region RegionOne monasca public http://157.159.232.245:8082/v2.0

openstack endpoint create --region RegionOne monasca internal http://157.159.232.245:8082/v2.0

openstack endpoint create --region RegionOne monasca admin http://157.159.232.245:8082/v2.0

openstack endpoint create --region RegionOne monasca --publicurl http://157.159.232.245:8082/v2.0 --internalurl http://157.159.232.245:8082/v2.0 --adminurl http://157.159.232.245:8082/v2.0

192.168.1.143 is the floating ip of my api vm address, change it to yours.  
create monasca api startup scripts,vim /etc/init/monasca-api.conf

# Startup script for the Monasca API

description "Monasca API Python app"

start on runlevel [2345]

console log

respawn

setgid monasca

setuid monasca

exec /usr/local/bin/gunicorn -n monasca-api -k eventlet --worker-connections=2000 --backlog=1000 --paste /etc/monasca/api-config.ini

start monasca-api service

service monasca-api start

if you get mysql connection error, modify monasca-common python file, and restart monasca-api service, the python code has bug reading mysql configuration. here is a quick hack,  
vim /usr/local/lib/python2.7/dist-packages/monasca\_common/repositories/mysql/mysql\_repository.py

self.conf = cfg.CONF

#self.database\_name = self.conf.mysql.database\_name

#self.database\_server = self.conf.mysql.hostname

#self.database\_uid = self.conf.mysql.username

#self.database\_pwd = self.conf.mysql.password

self.database\_name = 'mon'

self.database\_server = 'localhost'

self.database\_uid = 'monasca'

self.database\_pwd = 'qydcos'

install monasca-persister  
the monasca-persister java code has a bug writting data into influxdb, i fixed it and rebuild a jar file and upload it into monasca.git. the monasca-persister python code also has a bug writting data into influxdb, i have no time to fix it.

copy monasca-persister.jar file into /opt/monasca/  
copy persister-config.yml into /etc/monasca/

create monasca-persister startup script  
vim /etc/init/monasca-persister.conf

# Startup script for the Monasca Persister

description "Monasca Persister Python app"

start on runlevel [2345]

console log

respawn

setgid monasca

setuid monasca

exec /usr/bin/java -Dfile.encoding=UTF-8 -cp /opt/monasca/monasca-persister.jar monasca.persister.PersisterApplication server /etc/monasca/persister-config.yml

start monasca-persister

service monasca-persister start

install monasca-notificatoin

sudo pip install --upgrade monasca-notification

sudo apt-get install sendmail

copy notification.yaml into /etc/monasca/  
create startup script, vim /etc/init/monasca-notification.conf

# Startup script for the monasca\_notification

description "Monasca Notification daemon"

start on runlevel [2345]

console log

respawn

setgid monasca

setuid monasca

exec /usr/bin/python /usr/local/bin/monasca-notification

start notification service

service monasca-notification start

install monasca-thresh  
copy monasca-thresh into /etc/init.d/  
copy monasca-thresh.jar into /opt/monasca-thresh/  
copy thresh-config.yml into /etc/monasca/ and modify host, database to yours.  
start monasca-thresh

service monasca-thresh start

11, install monasca-agent  
install monasca-agent on openstack controller node, so that it can monitor openstack service process.

sudo pip install libxml2

sudo pip install --upgrade monasca-agent

setup monasca-agent, if you are on liberty, change user domain id and project domain id to default, for mitaka, use default domain id,

monasca-setup -u monasca -p qydcos --user\_domain\_id e25e0413a70c41449d2ccc2578deb1e4 --project\_domain\_id e25e0413a70c41449d2ccc2578deb1e4 --user monasca \

--project\_name service -s monitoring --keystone\_url http://192.168.1.11:35357/v3 --monasca\_url http://192.168.1.143:8082/v2.0 --config\_dir /etc/monasca/agent --log\_dir /var/log/monasca/agent --overwrite

source admin-rc.sh, run monasca metric-list

# monasca metric-list

monasca-setup -u monasca -p password -–user\_domain\_id \ a991f030143c488f84392cb905c38cf7 -–project\_domain\_id \ 02e76caff82c44c396179370f1562f9a --user monasca --project\_name service -s \ monitoring --keystone\_url [http://157.159.232.245:35357/v3](http://157.159.232.243:35357/v3) –monasca\_url \ http://157.159.232.245:8082/v2.0 --config\_dir /etc/monasca/agent –log\_dir \ /var/log/monasca/agent --overwrite

| Observation(Error) | Hypothesis(Remark) | Experimentation(Action) |
| --- | --- | --- |
| service monasca-thresh start  Error |  |  |
| monasca metric-list  Failed to connect to http://157.159.232.245:8070/v2.0, error was HTTPConnectionPool(host='157.159.232.245', port=8070): Max retries exceeded with url: /v2.0/metrics (Caused by NewConnectionError('<requests.packages.urllib3.connection.HTTPConnection object at 0x7fdd28060d10>: Failed to establish a new connection: [Errno 111] Connection refused',)) | Devstack miss configured | Rebuild devstack changing monasc\_api port to from 8070 to 8082 |
| Failed to connect to http://10.11.12.5:8082/v2.0, error was HTTPConnectionPool(host='10.11.12.5', port=8082): Max retries exceeded with url: /v2.0/metrics (Caused by NewConnectionError('<requests.packages.urllib3.connection.HTTPConnection object at 0x7efd8bea3cd0>: Failed to establish a new connection: [Errno 111] Connection refused',)) |  |  |
| telnet localhost 8082  Trying 127.0.0.1...  telnet: Unable to connect to remote host: Connection refused |  |  |
| sudo /usr/local/bin/gunicorn -n monasca-api -k eventlet --backlog=1000 --paste /etc/monasca/api-config.ini  ImportError: No module named middleware.keystone\_context\_filter | I have installed all these packages I am still getting the error message. I am using django 1.7. Following is my freeze list: defusedxml==0.4.1 Django==1.7 django-cors-headers==1.3.1 django-tastypie==0.13.3 lxml==3.6.4 oauth2-provider==0.0 Pillow==3.4.2 python-dateutil==2.6.0 python-mimeparse==1.6.0 six==1.10.0 South==1.0.2 |  |
| sudo pip install falcon  sudo apt-get install python-oslo-log  sudo apt-get install python-keystonemiddleware  sudo apt-get install python-oslo-middleware  sudo pip install simplejson  sudo pip install voluptuous  sudo pip install kafka  sudo pip install influxdb  sudo pip install pyparsing  sudo pip install pymysql  sudo pip install validate\_email  Make sure paste.deploy is installed to have gunicorn to run. Use the command sudo pip install PasteDeploy, and eventlet with sudo pip install eventlet | gunicorn -k eventlet --worker-connections=2000 --backlog=1000 --paste /etc/monasca/api-config.ini --log-file=- |  |
|  |  |  |
|  |  |  |
| what’s the result of changing in File “/usr/local/lib/python2.7/dist-packages/monasca\_api/api/server.py”, line 120, in launch dimension\_names = simport.load(cfg.CONF.dispatcher.dimension\_names)() to dimension\_names = simport.load(‘monasca\_api.v2.reference.metrics:DimensionValues’)() |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| ERROR gunicorn.error RepositoryException: Failed to import 'monasca\_api.common.repositories.mysql.alarm\_definitions\_repository'. Error: No module named mysql.alarm\_definitions\_repository |  |  |

Check : <https://github.com/BU-NU-CLOUD-SP16/MOC-Monitoring/wiki/How-to-Install-Monasca-API>