# statsd搭建

## statsd安装

1. 安装nodejs

|  |
| --- |
| yum install -y nodejs |

1. 下载statsd软件包：

|  |
| --- |
| wget -O /app/statsd-0.8.0.tar.gz https://github.com/etsy/statsd/archive/v0.8.0.tar.gz |

1. 解压：

|  |
| --- |
| cd /app;tar -xzf statsd-0.8.0.tar.gz |

1. 安装依赖包

|  |
| --- |
| cd statsd-0.8.0  npm install  npm install kafka-node -S |

1. 验证statsd是否可以运行

|  |
| --- |
| ./run\_tests.sh |

显示如下则为成功吗，如不成功，请删除目录下的node\_modules目录，检查网络环境然后重新安装依赖（步骤4）：



## 配置statsd

1. 创建/etc/statsd/statsd.js配置文件，内容如下：

|  |
| --- |
| {  flushInterval: 300000,//刷新时间5min  //将zookeeper1、zookeeper3、zookeeper3改为实际的zookeeper服务器IP地址  zookeeper: 'zookeeper1:2181, zookeeper2:2181, zookeeper3:2181',  clientId: 'statsd',  zkOptions: {sessionTimeout: 300,spinDelay: 100,retries: 2},  kafkaTopic: 'sfo-statsd-info',  backends: [ "./backends/kafkaBackend" ],  } |

1. 创建/app/statsd-0.8.0/backends/kafkaBackend.js配置文件，内容如下：

|  |
| --- |
| var util = require('util');  var kafka = require('kafka-node');  var Producer;  var client;  var producer;  var kafkaTopic;  var clientId;  var zkOptions;  function kafkaBackend(startupTime, config, emitter){  var self = this;  this.lastFlush = startupTime;  this.lastException = startupTime;  this.zookeeper = config.zookeeper || '';  this.clientId = config.clientId;  this.zkOptions = config.zkOptions;  this.kafkaTopic =config.kafkaTopic;  Producer = kafka.HighLevelProducer;  client = new kafka.Client(this.zookeeper, this.clientId, this.zkOptions);  producer = new Producer(client);  producer.on('ready', function () {  console.log('producer is ready')  });  emitter.on('flush', function(timestamp, metrics) { self.flush(timestamp, metrics); });  emitter.on('status', function(callback) { self.status(callback); });  }  kafkaBackend.prototype.flush = function(timestamp, metrics) {  console.log('Flushing stats at ', new Date(timestamp \* 1000).toString());  var out = {  counters: metrics.counters,  timers: {},  gauges: metrics.gauges,  timer\_data: metrics.timer\_data,  counter\_rates: metrics.counter\_rates,  sets: function (vals) {  var ret = {};  for (var val in vals) {  ret[val] = vals[val].values();  }  return ret;  }(metrics.sets),  pctThreshold: metrics.pctThreshold,  timestamp: timestamp \* 1000,  };  var payloads = [  {  topic: this.kafkaTopic,  messages: JSON.stringify(out)  },  ];  producer.send(payloads, function (err, data) {  console.log(data || err);  });  producer.on('error', function (err) {  console.log('!!!!!!!!!!!!!!!!!!!'+err)  })  };  kafkaBackend.prototype.status = function(write) {  ['lastFlush', 'lastException'].forEach(function(key) {  write(null, 'console', key, this[key]);  }, this);  };  exports.init = function(startupTime, config, events) {  var instance = new kafkaBackend(startupTime, config, events);  return true;  }; |

## 使用systemd管理statsd服务

1. 创建 /usr/lib/systemd/system/statsd.service 文件，内容如下：

|  |
| --- |
| [Unit] Description=statsd service Wants=network-online.target After=syslog.target network-online.target  [Service]  ExecStart=/usr/bin/node /app/statsd-0.8.0/stats.js /etc/statsd/statsd.js  Restart=always User=nobody Group=nobody StandardOutput=syslog  StandardError=syslog SyslogIdentifier=statsd.service Environment=PATH=/usr/bin:/usr/local/bin Environment=NODE\_ENV=production WorkingDirectory=/app/statsd-0.8.0/      [Install] WantedBy=multi-user.target |

1. 加载服务

|  |
| --- |
| systemctl daemon-reload |

1. 启动服务

|  |
| --- |
| systemctl start statsd.service |

1. 查看日志

|  |
| --- |
| journalctl -u statsd.service |

注：statsd服务的日志默认输出到/var/log/message，可以通过配置rsyslog等相关文件改变日志的输出文件，以下提供一种方法：

1. 创建 /etc/rsyslog.d/statsd.conf，内容如下：

|  |
| --- |
| if $programname == 'statsd.service' then /var/log/statsd.log  & stop |

1. 重启rsyslog服务：：

|  |
| --- |
| systemctl restart rsyslog.service |

## swift接入statsd

swift接入statsd很简单，只需在每个server的配置文件（proxy-serve.conf、account-server.conf、container-server.conf、object-server.conf）的[DEFAULT]部分加入相关配置，重启相关服务即可：

|  |
| --- |
| [DEFAULT]  …..  log\_statsd\_host = statsd\_server\_ip  log\_statsd\_port = 8125  log\_statsd\_default\_sample\_rate = 1.0  log\_statsd\_sample\_rate\_factor = 1.0  log\_statsd\_metric\_prefix = 格式为节点角色+主机名，如：proxy\_server\_node1  ….. |
|  |