

Twerk Media RTB Administration and Analytics Server

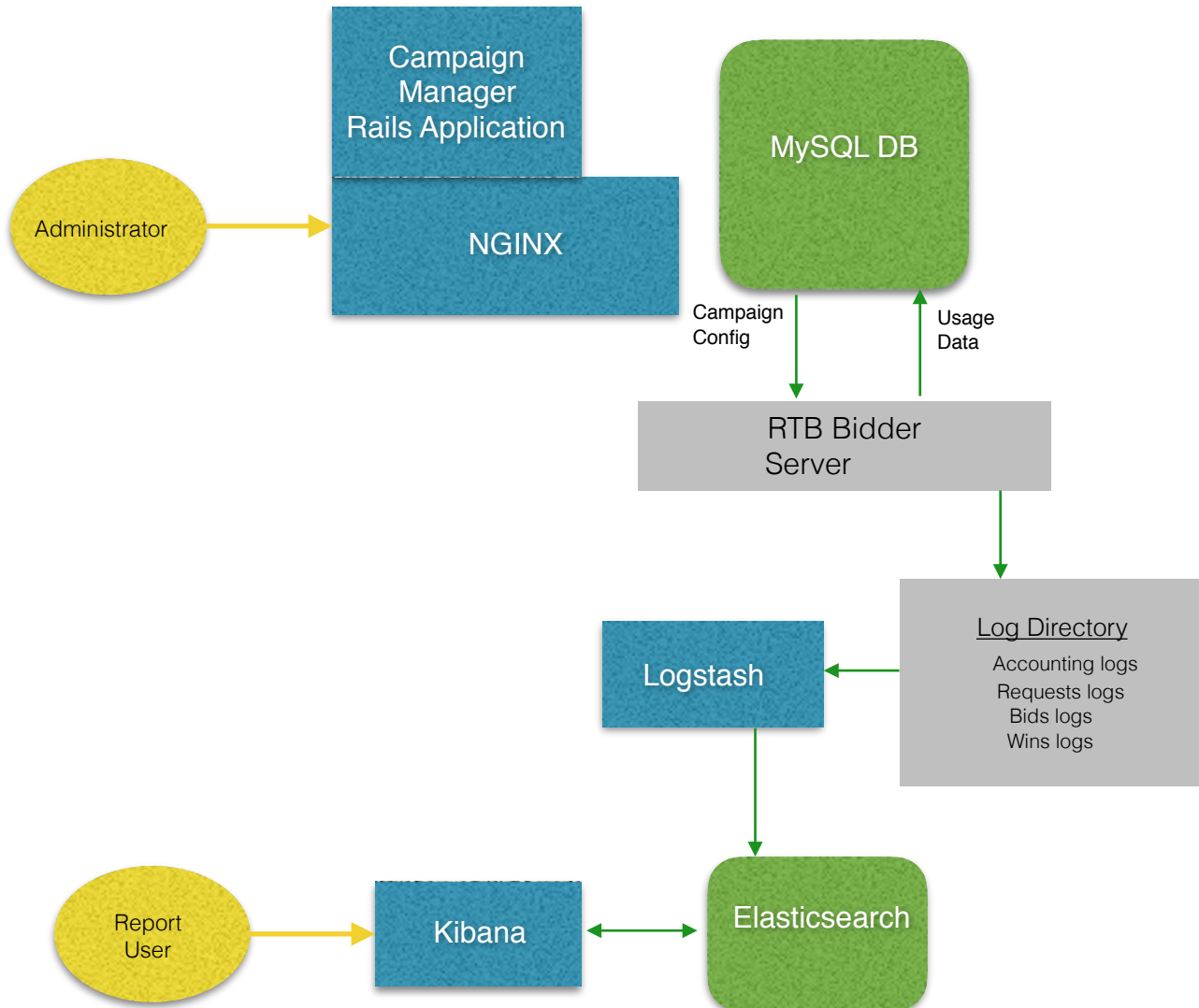
November 18, 2016

Overview

This document describes the administration and report server for the Real Time Bidder implemented for Twerk Media. This includes the following:

1. Campaign Manager. A web application which allows users to create campaigns used by the RTB bidder. This consists of the following components:
 1. Ruby on Rails application - defines the web application.
 2. Nginx web server - the Rails application runs on top of this web server.
 3. MySQL - database that contains the web application and RTB usage data. This database is updated by the web application and the RTB bidder via the Crosstalk application.
 2. Reporting System. A data store and analytics reporting system for detailed RTB records. This consists of the following components.
 1. Elasticsearch. Database that stores all data.
 2. Logstash. Reads log files placed in a log directory, then translates and inserts records into the Elasticsearch database.
 3. Kibana. User interface and web application for querying and creating visualizations from Elasticsearch data.
-

Architecture



Access

The following credentials were used to access the server.

- Applications and packages were installed using system user id `rtb4free` with password `"AB~mM698sZy^"` (without quotes). This id has sudo access.
 - Mysql installation
 - root password is set to `AB~mM698sZy^`
 - Web application and Bidder access is accessed using id `ben` with password `whatweneednowislove`
 - The Campaign Manager web application is accessed at URL `http://195.154.154.177/`
 - An administrator login has been created for kate@twerkmedia.com. The administrator can create additional logins via the web application.
 - The Kibana web application is accessed at <http://195.154.154.177:5601/>
 - NOTE!!! Kibana does not have a log in front end or password protection. You must restrict access using other methods. If you need log in access, you can consider purchasing support service from Elastic which includes X-Pack Security. (see <https://www.elastic.co/guide/en/x-pack/current/xpack-introduction.html>)
-

Installation

The following components were installed

1. Rails

Ruby installed under id rtb4free using rvm.

Rails app installed in directory /home/rtb4free/rtb4free_admin.

Reference: <https://www.digitalocean.com/community/tutorials/how-to-install-ruby-on-rails-on-ubuntu-14-04-using-rvm>

2. Mysql. Already installed.

3. Nginx. Installed with Phusion support to run rails.

Reference <https://www.digitalocean.com/community/tutorials/how-to-install-rails-and-nginx-with-passenger-on-ubuntu>

Configuration files in /opt/nginx/conf/nginx.conf

4. Java 8. Required for Elasticsearch

Reference <http://tecadmin.net/install-java-8-on-debian/#>

5. Elasticsearch. Installed using apt.

Reference <https://www.elastic.co/guide/en/elasticsearch/reference/5.0/deb.html>

Configuration files in /etc/elasticsearch/elasticsearch.yml

After new Elasticsearch install, execute commands to load dynamic templates - see appendix.

6. Kibana. Installed using apt.

Reference <https://www.elastic.co/guide/en/elasticsearch/reference/5.0/deb.html>

Configuration files in /etc/kibana/kibana.yml

7. Logstash. Installed using apt.

Reference <https://www.elastic.co/guide/en/elasticsearch/reference/5.0/deb.html>

Configuration files in /etc/logstash/logstash.yml.

Input configuration definitions in directory /etc/logstash/conf.d/. See appendix.

Operations

The following are commands for starting and stopping components.

Campaign Manager

Controlling nginx will control the rails app.

```
sudo systemctl start nginx.service  
sudo systemctl stop nginx.service
```

Mysql

```
sudo systemctl start mysql.service  
sudo systemctl stop mysql.service
```

Elasticsearch

```
sudo service elasticsearch start  
sudo service elasticsearch stop
```

Kibana

```
sudo systemctl start kibana.service  
sudo systemctl stop kibana.service
```

Logstash

```
sudo systemctl start logstash.service  
sudo systemctl stop logstash.service
```

Appendix - Logstash Configuration Files

These files map the RTB Files to Elasticsearch document formats.

accounting.conf

```
input {
  file {
    path => "/home/rtb4free/XRTB/data/twerkmedia/logstash/
accounting*"
    type => "accounting"
    start_position => "beginning"
    codec => "json"
  }
}
filter {
  if [type] == "accounting" {
    date {
      match => ["time", "UNIX_MS"]
    }
  }
}
output {
  if [type] == "accounting" {
    elasticsearch {
      hosts => ["localhost:9200"]
      index => "rtbaccounting-%{+YYYY.MM.dd}"
    }
  }
}
# stdout { codec => rubydebug }
}
```

bids.conf

```
input {
  file {
    path => "/home/rtb4free/XRTB/data/twerkmedia/logstash/bids*"
    type => "bids"
    start_position => "beginning"
    codec => "json"
  }
}
filter {
  if [type] == "bids" {
    date {
      match => ["utc", "UNIX_MS"]
    }
    if [lat] and [lon] {
      mutate {
        add_field => {
          "location" => "%{lat},%{lon}"
        }
      }
    }
  }
}
output {
  if [type] == "bids" {
    elasticsearch {
      hosts => ["localhost:9200"]
      index => "rtbbids-%{+YYYY.MM.dd}"
      document_id => "%{oidStr}"
    }
  }
}
# stdout { codec => rubydebug }
}
```


wins.conf

```
input {
  file {
    path => "/home/rtb4free/XRTB/data/twerkmedia/logstash/wins*"
    type => "wins"
    start_position => "beginning"
    codec => "json"
  }
}
filter {
  if [type] == "wins" {
    date {
      match => ["utc", "UNIX_MS"]
    }
    if [lat] and [lon] {
      mutate {
        add_field => {
          "location" => "%{lat},%{lon}"
        }
      }
    }
  }
}
output {
  if [type] == "wins" {
    elasticsearch {
      hosts => ["localhost:9200"]
      index => "rtbwins-%{+YYYY.MM.dd}"
      document_id => "%{hash}"
    }
  }
}
# stdout { codec => rubydebug }
}
```

requests.conf

```
input {
  file {
    path => "/home/rtb4free/XRTB/data/twerkmedia/logstash/
request*"
    type => "requests"
    start_position => "beginning"
    codec => "json"
  }
}
filter {
  if [type] == "requests" {
    date {
      match => ["[ext][timestamp]", "UNIX_MS"]
    }
    if [device][geo][lat] and [device][geo][lon] {
      mutate {
        add_field => {
          "[device][geo][location]" => "%{[device][geo][lat]},%{[device]
[geo][lon]}"
        }
      }
    }
  }
}
output {
  if [type] == "requests" {
    elasticsearch {
      hosts => ["localhost:9200"]
      index => "rtbrequests-%{+YYYY.MM.dd}"
      document_id => "%{id}"
    }
  }
}
# stdout { codec => rubydebug }
}
```

Appendix - Elasticsearch Template Files

These files map the log entries to Elasticsearch fields

```
curl -X PUT http://127.0.0.1:9200/_template/rtbaccounting_template -d
{
  "template": "rtbaccounting*",
  "settings": {
    "number_of_shards": 5
  },
  "mappings": {
    "accounting": {
      "properties": {
        "time": {
          "type": "date",
          "format": "epoch_millis"
        },
        "winPrice": {
          "type": "scaled_float",
          "scaling_factor": 100000
        },
        "bidPrice": {
          "type": "scaled_float",
          "scaling_factor": 100000
        }
      }
    }
  }
}
```

```
curl -X PUT http://127.0.0.1:9200/_template/rtbwin_template -d '{
  "template": "rtbwin*",
  "settings": {
    "number_of_shards": 5
  },
  "mappings": {
    "wins": {
      "properties": {
        "utc": {
          "type": "date",
          "format": "epoch_millis"
        },
        "location": {
          "type": "geo_point"
        },
        "price": {
          "type": "scaled_float",
          "scaling_factor": 100000
        },
        "cost": {
          "type": "scaled_float",
          "scaling_factor": 100000
        }
      }
    }
  }
}
```

```
curl -X PUT http://127.0.0.1:9200/_template/rtbbid_template -d '{
  "template": "rtbbid*",
  "settings": {
    "number_of_shards": 5
  },
  "mappings": {
    "bids": {
      "properties": {
        "utc": {
          "type": "date",
          "format": "epoch_millis"
        },
        "location": {
          "type": "geo_point"
        },
        "price": {
          "type": "scaled_float",
          "scaling_factor": 100000
        },
        "cost": {
          "type": "scaled_float",
          "scaling_factor": 100000
        }
      }
    }
  }
}
```

```
curl -X PUT http://127.0.0.1:9200/_template/rtbrequest_template -d '{
  "template": "rtbrequest*",
  "settings": {
    "number_of_shards": 5
  },
  "mappings": {
    "requests": {
      "properties": {
        "device": {
          "type": "object",
          "properties": {
            "geo": {
              "type": "object",
              "properties": {
                "location": {
                  "type": "geo_point"
                }
              }
            }
          }
        }
      },
      "ext": {
        "type": "object",
        "properties": {
          "timestamp": {
            "type": "date",
            "format": "epoch_millis"
          }
        }
      }
    }
  }
}
```
