Redash HTTPS (SSL) Setup

Prerequisite: VM should have a public IP Address, a domain name to be associated with that Public IP, and port 80, 443 to be opened

Installing Certificates

- Login to Redash VM
  ```
  sudo yum install -y epel-release
  sudo yum install -y yum-utils
  sudo yum install -y certbot
  ```
- To Install certificates
  ```
  sudo certbot certonly --standalone
  ```
  Provide group gmail ID(alerts-saas-prod1@argoid.com) and the domain name in certbot prompt
- Certs will be installed in **/etc/letsencrypt/live/*domain_name*/**

```
[manjunath@argoid-saas-prod1-host-056 ~]$ sudo ls /etc/letsencrypt/live/analytics.prod.rarerabbit.saas.argoid.com/
cert.pem  chain.pem  fullchain.pem  privkey.pem  README
[manjunath@argoid-saas-prod1-host-056 ~]$
```

Configuring SSL in Redash

- Create Nginx configuration file
  ```
  vi /opt/redash/nginx/nginx.conf
  ```

```
upstream redash {
    server redash:5000;
}

server {
    listen      80;
    listen [::]:80;
    server_name  analytics.prod.rarerabbit.saas.argoid.com;

    location ^~ /ping {
        proxy_set_header Host $http_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_pass       http://redash;
    }

    location / {
        rewrite ^ https://$host$request_uri? permanent;
    }

    location ^~ /.well-known {
        allow all;
        root  /data/letsencrypt/;
    }
}

server {
```

```
 listen      443           ssl http2;
 listen [::]:443           ssl http2;
 server_name               analytics.prod.rarerabbit.saas.argoid.com;

 add_header                Strict-Transport-Security "max-
age=31536000" always;

 ssl_session_cache         shared:SSL:20m;
 ssl_session_timeout       10m;

 ssl_protocols             TLSv1 TLSv1.1 TLSv1.2;
 ssl_prefer_server_ciphers on;
 ssl_ciphers               "ECDH+AESGCM:ECDH+AES256:ECDH+AES128:!ADH:!
AECDH:!MD5;";

 ssl_stapling              on;
 ssl_stapling_verify       on;
 resolver                  8.8.8.8 8.8.4.4;

 ssl_certificate           /etc/letsencrypt/live/analytics.prod.
rarerabbit.saas.argoid.com/fullchain.pem;
 ssl_certificate_key       /etc/letsencrypt/live/analytics.prod.
rarerabbit.saas.argoid.com/privkey.pem;
 ssl_trusted_certificate   /etc/letsencrypt/live/analytics.prod.
rarerabbit.saas.argoid.com/chain.pem;

 access_log                /dev/stdout;
 error_log                 /dev/stderr info;

 # other configs

 location / {
     proxy_set_header Host $http_host;
     proxy_set_header X-Real-IP $remote_addr;
     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
     proxy_set_header X-Forwarded-Proto $scheme;

     proxy_pass       http://redash;
 }
}
```

Mention domain name in `server_name` sections, also mention the correct path of certs file in `ssl_certificate` `ssl_certificate_key` `ssl_trusted_certificate` section (Certs file will be mounted inside the container, so the path of the certs file should start with /etc/letsencrypt/live/)

- Modify the docker-compose file to mount the certs inside Nginx `/opt/redash/docker-compose.yml`

```
    volumes:
      - /opt/redash/nginx/nginx.conf:/etc/nginx/conf.d/default.conf
      - /etc/letsencrypt/:/etc/letsencrypt/
```

Also, enable host port-mapping on Port 443

```
 nginx:
    image: redash/nginx:latest
    ports:
      - "80:80"
      - "443:443"
    depends_on:
      - server
    links:
      - server:redash
```

Fully functional docker-compose file should look like

```
version: "2"
x-redash-service: &redash-service
  image: redash/redash:8.0.0.b32245
  depends_on:
    - postgres
    - redis
  env_file: /opt/redash/env
  restart: always
services:
  server:
    <<: *redash-service
    command: server
    ports:
      - "5000:5000"
    environment:
      REDASH_WEB_WORKERS: 4
  scheduler:
    <<: *redash-service
    command: scheduler
    environment:
      QUEUES: "celery"
      WORKERS_COUNT: 1
  scheduled_worker:
    <<: *redash-service
    command: worker
    environment:
      QUEUES: "scheduled_queries,schemas"
```

```
      WORKERS_COUNT: 1
  adhoc_worker:
    <<: *redash-service
    command: worker
    environment:
      QUEUES: "queries"
      WORKERS_COUNT: 2
  redis:
    image: redis:5.0-alpine
    restart: always
  postgres:
    image: postgres:9.6-alpine
    env_file: /opt/redash/env
    volumes:
      - /opt/redash/postgres-data:/var/lib/postgresql/data
    restart: always
  nginx:
    image: redash/nginx:latest
    ports:
      - "80:80"
      - "443:443"
    depends_on:
      - server
    links:
      - server:redash
    volumes:
      - /opt/redash/nginx/nginx.conf:/etc/nginx/conf.d/default.conf
      - /etc/letsencrypt/:/etc/letsencrypt/
    restart: always
```

- Restart Redash containers

  ```
  cd /opt/redash/
  /usr/local/bin/docker-compose up -d
  ```
- docker-compose -f docker-compose.yml up -d.

Reference: https://gist.github.com/arikfr/64c9ff8d2f2b703d4e44fe9e45a7730e