Deploy using Docker

**Requirements**

Solution: As per the requirements listed in the document,

1. We would need to pull the images at our local system and perform customization.
2. The dependent services are Redis server and PostgreSQL: We can pull the required images first from the registry or directly run the container:

docker run –d –name redis\_sentry redis:3.2 alpine – to startup the redis server

docker run –d –name postgres\_sentry –e POSTGRES\_PASSWORD=secret –e POSTGRES\_USER=sentry postgres:9.6

1. Generate secret key –

docker run --rm sentry config generate-secret-key(let the generated secret key be 123abc)

1. Starting sentry server –

docker run –d –name sentry\_server –e SENTRY\_SECRET\_KEY=’123abc’ -p 9000:9000 –link redis\_sentry:redis –link postgres\_sentry:postgres sentry

1. Configure the initial user

docker run –it –rm –e SENTRY\_SECRET\_KEY=’123abc’ –link redis\_sentry:redis –link postgres\_sentry:postgres sentry createuser

1. Also, the default config needs a default set of background worker

docker run -d --name sentry-cron -e SENTRY\_SECRET\_KEY=’123abc’ --link postgres\_sentry:postgres --link redis:redis sentry run cron

docker run -d --name sentry-worker-1 -e SENTRY\_SECRET\_KEY=’123abc’ --link postgres\_sentry:postgres --link redis:redis sentry run worker

1. Sentry cron process startup:

docker run –d –name cron\_sentry sentry-onpremise run cron