

Flask with Cloud watch(AWS)

Sno	Date	Modification	Author	Verified By
1	2019/08/22	Initial Document	Nishtha	Sumit Goyal

Table of Contents

python/flask with cloud watch(AWS) **Error! Bookmark not defined.**

Objective **Error! Bookmark not defined.**

Solutions:.....	3
Steps to connect flask with cloud watch.....	Error! Bookmark not defined.
Final Result.	Error! Bookmark not defined.

Python/flask With Cloud Watch(AWS)

Watchtower is a log handler for Amazon Web Services Cloud Watch Logs. Cloud Watch Logs is a log management service built into AWS. It is conceptually similar to services like Splunk and Loggly, but is more lightweight, cheaper, and tightly integrated with the rest of AWS.



Objective

The main objective of this project is that we will be able to generate log files and it will automatically appear in our Cloud watch and provides real time monitoring

Solutions:

Note:In this document we explain step by step Implementation between flask and Cloud watch (To store logs).

Step1 :

```
pipinstallwatchtower
```

Step2:

```
Pip install awscli
```

This will install the aws-cli package as well as all dependencies.

The quickest way to get started is to run the **aws configure** command:

Step3: Go to your flask project directory and run the following command for connect to aws cloud watch

```
$ aws configure

AWS Access Key ID: your aws access key

AWS Secret Access Key: your aws secret key

Default region name [us-west-2]: your aws region

Default output format [None]: json
```

Step4: Go to your main.py file and wirte the following code.

```
import watchtower, flask, logging

logging.basicConfig(level=logging.INFO)
app = flask.Flask("loggable")
handler = watchtower.CloudWatchLogHandler()
app.logger.addHandler(handler)
logging.getLogger("werkzeug").add Handler(handler)
```

After that run your flask project by entering the command

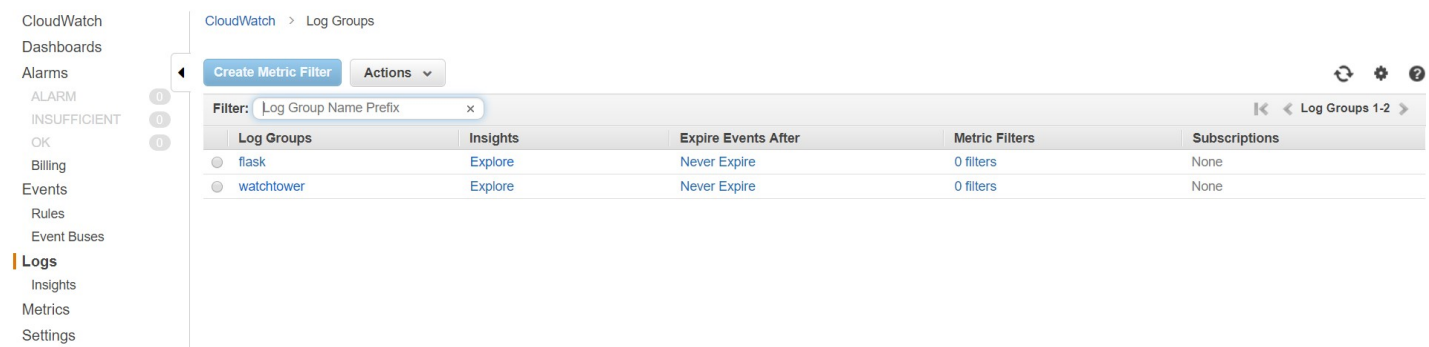


When you hit this command in the command prompt the following screen will display

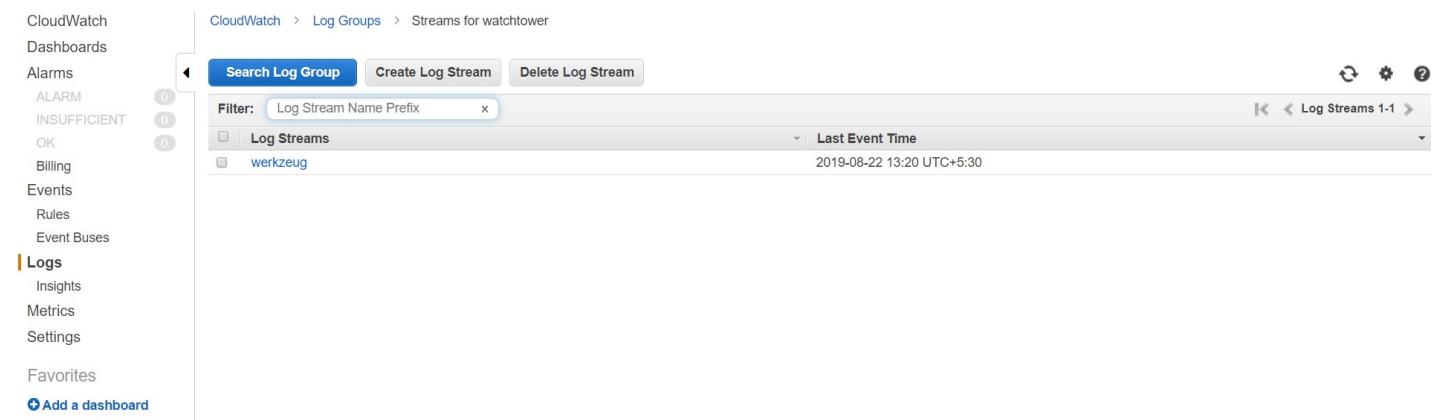
```
F:\flask_app>python run.py
C:\Python\lib\site-packages\flask_sqlalchemy\__init__.py:835: FSADeprecationWarning: SQLAlchemy_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
  "SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and
INFO:botocore.credentials:Found credentials in shared credentials file: ~/.aws/credentials
* Serving Flask app "flaskblog" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
INFO:werkzeug: * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:19:35] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:19:35] "GET /static/main.css HTTP/1.1" 304 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:19:36] "GET /favicon.ico HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:19:42] "GET /jb HTTP/1.1" 404 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:05] "GET / HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:07] "GET /login HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:08] "GET /register HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:09] "GET /about HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:10] "GET /home HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:12] "GET /login HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:14] "GET /register HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [22/Aug/2019 13:20:18] "GET /register/snow HTTP/1.1" 404 -
```

Step5: Now go to your AWS console

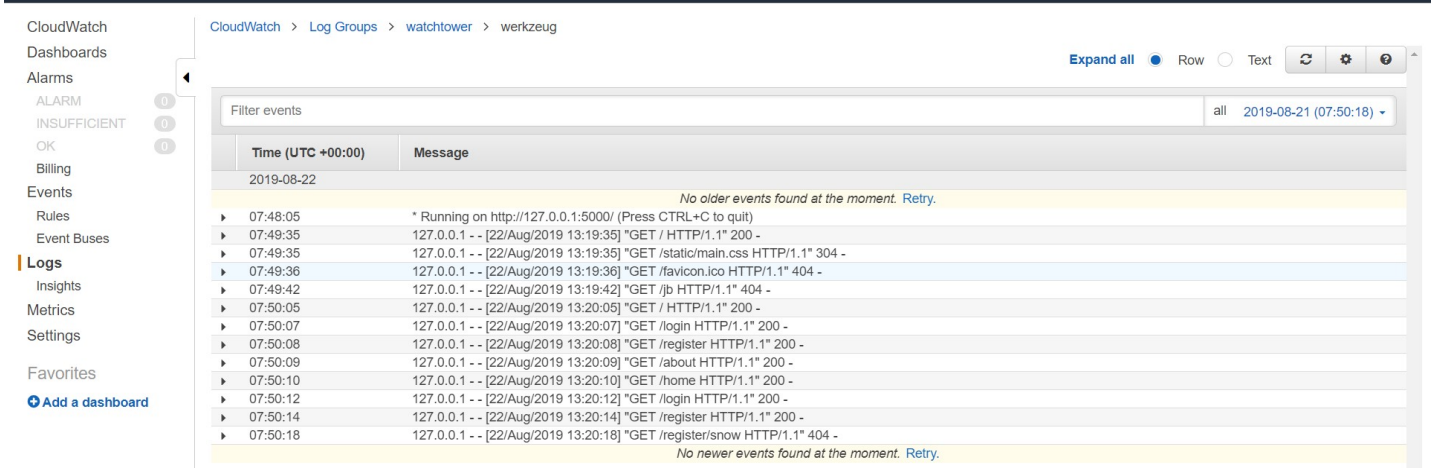
Click on the logs from the side navbar. you will see the following screen.



When you click on the watchtower button. You will see your log file name. Whatever name you defined in your flask app.. in my app I defined name werkzeug.



When you click on your log file you will see all the log file generated from your flask app. your screen looks something like this.



when you click on the cloud watch you may see now you are able to see all your flask app logs in real-time monitoring.

