Singapore Travel Guide

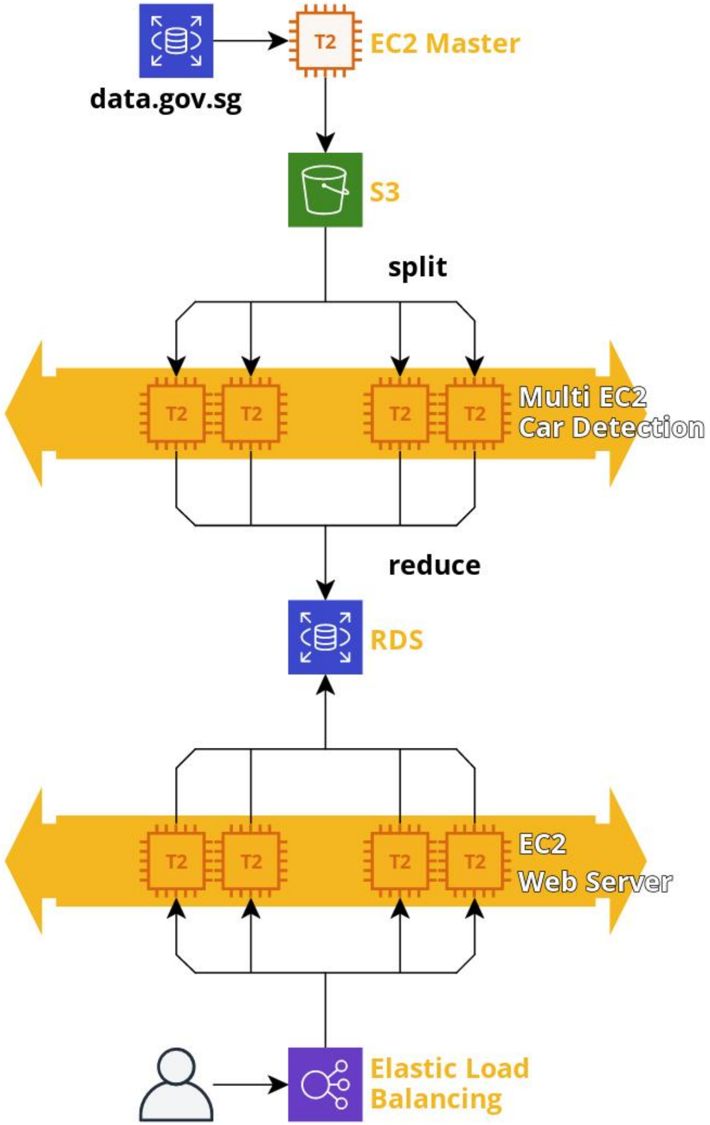
Project Milestone #2

NUS
National University
of Singapore

Outline

- Overview
- UI Design Diagram
- Image Processing
  - Car Detection
  - SQL
- Cloud Architecture
  - Web Server On Cloud
  - Image Processing On Cloud
  - Elasticity & Scalability
- Implementation Plan

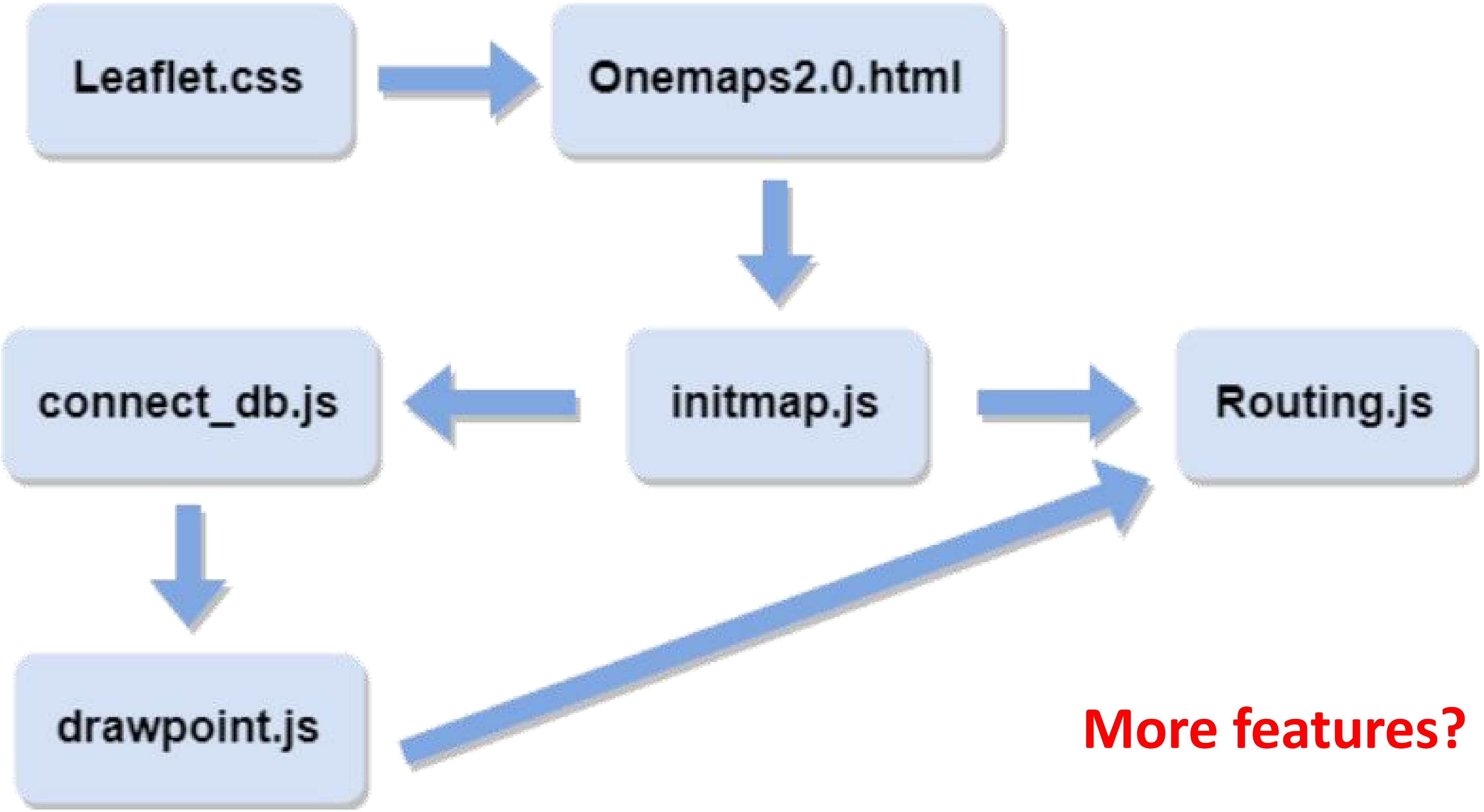
Overview


data.gov.sg
T2
EC2 Master
S3
split
T2
Multi EC2
Car Detection
reduce
RDS
T2
EC2
Web Server
Elastic Load
Balancing

Outline

- Overview
- UI Design Diagram
- Image Processing
  - Car Detection
  - SQL
- Cloud Architecture
  - Web Server On Cloud
  - Image Processing On Cloud
  - Elasticity & Scalability
- Implementation Plan

UI Design Diagram

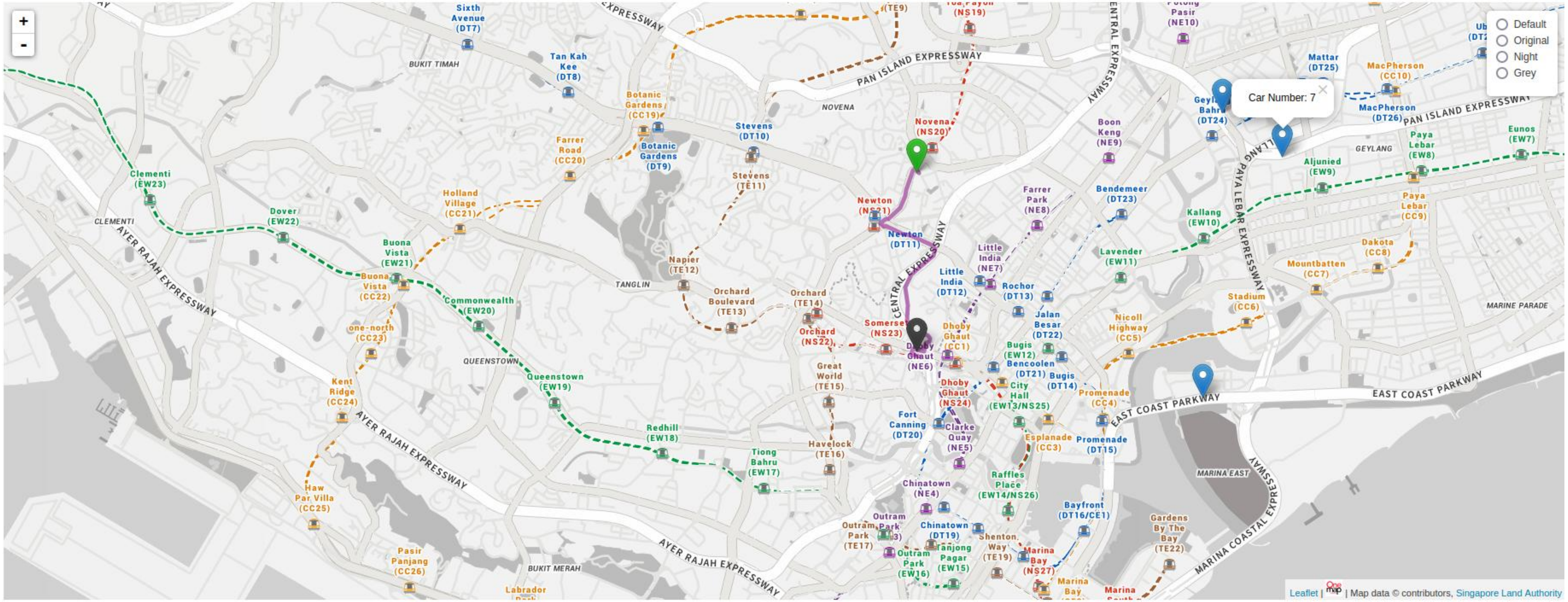


More features?

- When the user enters the start and end points, Routing.js draws the line on the map
- To show all the the location of all cameras on the map, as well as the number of vehicles

UI Design Diagram

Singapore Map


+
-
○ Default
○ Original
○ Night
○ Grey
Car Number: 7
Leaflet | OneMap | Map data © contributors, Singapore Land Authority

Outline

- Overview
- UI Design Diagram
- Image Processing
    - Car Detection
    - SQL
- Cloud Architecture
    - Web Server On Cloud
    - Image Processing On Cloud
    - Elasticity & Scalability
- Implementation Plan

Image Processing

- **Car Detection**
- We will use machine learning to count the num of cars

- We use anchor boxes to mark cars and count the num of anchor boxes

- The program runs every 5 minutes to fetch the data from data.gov.sg and process it in AWS EC2

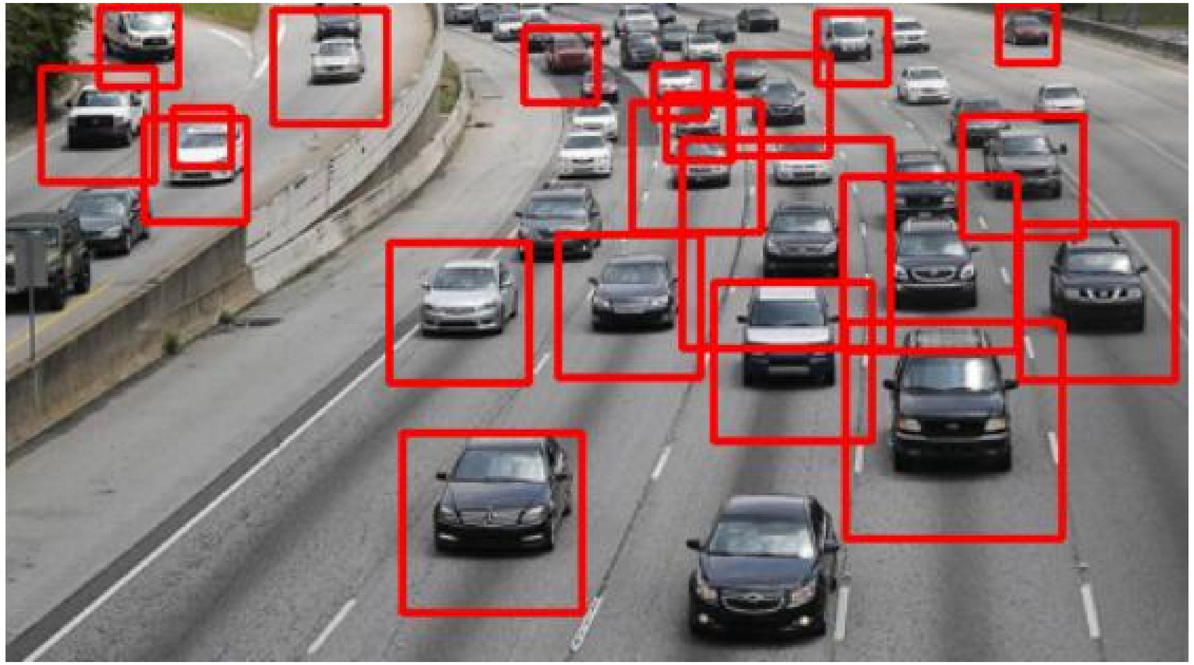
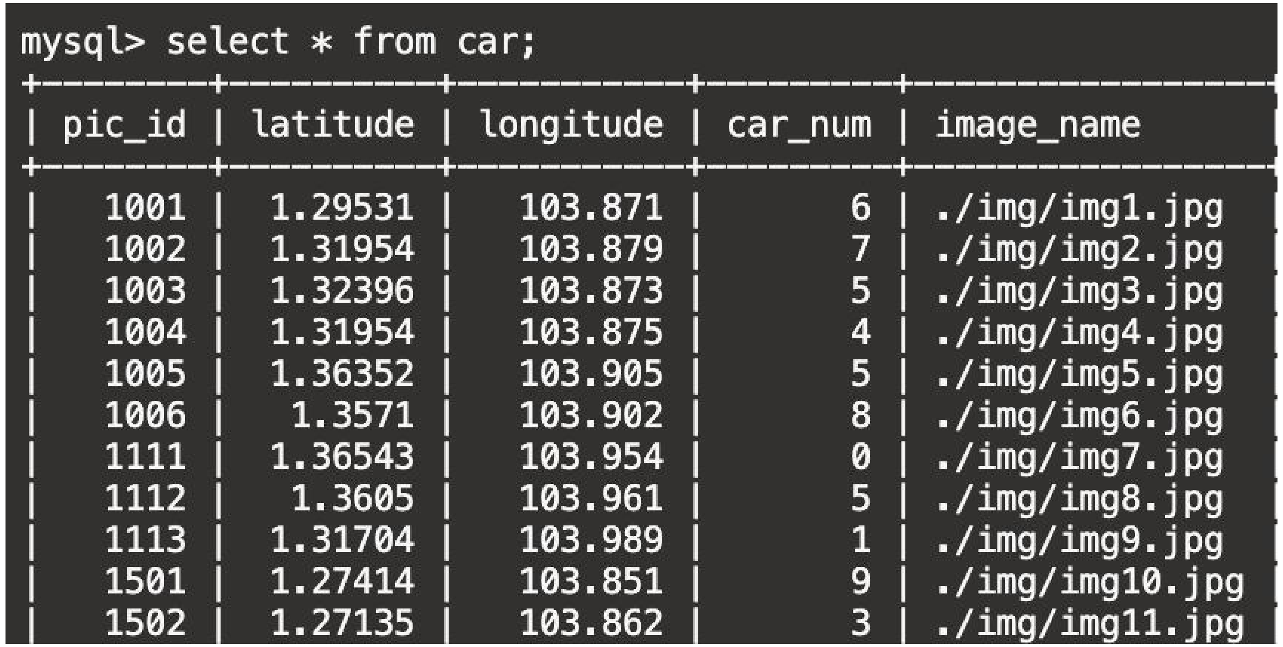


Image Processing

- **SQL**
- We use AWS RDS to store the result.
- The front-end will get the information from SQL and print it in the map.
- For each camera , we have its camera id as its primary key and location to mark it on the map.



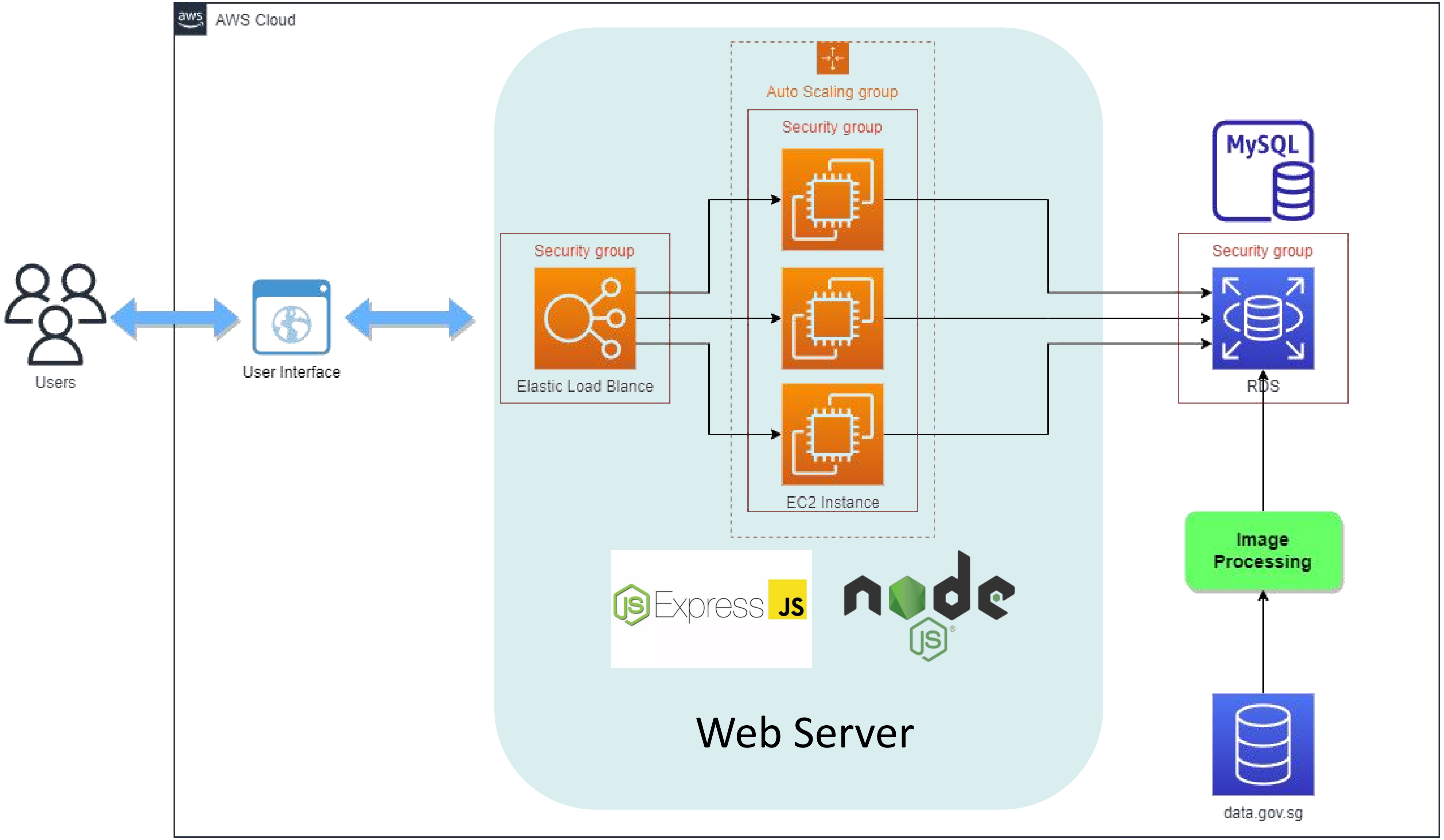
```
mysql> select * from car;
+--------+----------+-----------+---------+-----------------
| pic_id | latitude | longitude | car_num | image_name
+--------+----------+-----------+---------+-----------------
|   1001 |  1.29531 |   103.871 |       6 | ./img/img1.jpg
|   1002 |  1.31954 |   103.879 |       7 | ./img/img2.jpg
|   1003 |  1.32396 |   103.873 |       5 | ./img/img3.jpg
|   1004 |  1.31954 |   103.875 |       4 | ./img/img4.jpg
|   1005 |  1.36352 |   103.905 |       5 | ./img/img5.jpg
|   1006 |   1.3571 |   103.902 |       8 | ./img/img6.jpg
|   1111 |  1.36543 |   103.954 |       0 | ./img/img7.jpg
|   1112 |   1.3605 |   103.961 |       5 | ./img/img8.jpg
|   1113 |  1.31704 |   103.989 |       1 | ./img/img9.jpg
|   1501 |  1.27414 |   103.851 |       9 | ./img/img10.jpg
|   1502 |  1.27135 |   103.862 |       3 | ./img/img11.jpg
```

Outline

- Overview
- UI Design Diagram
- Image Processing
  - Car Detection
  - SQL
- Cloud Architecture
  - Web Server On Cloud
  - Image Processing On Cloud
  - Elasticity & Scalability
- Implementation Plan

Cloud Architecture

- **Web Server On Cloud**

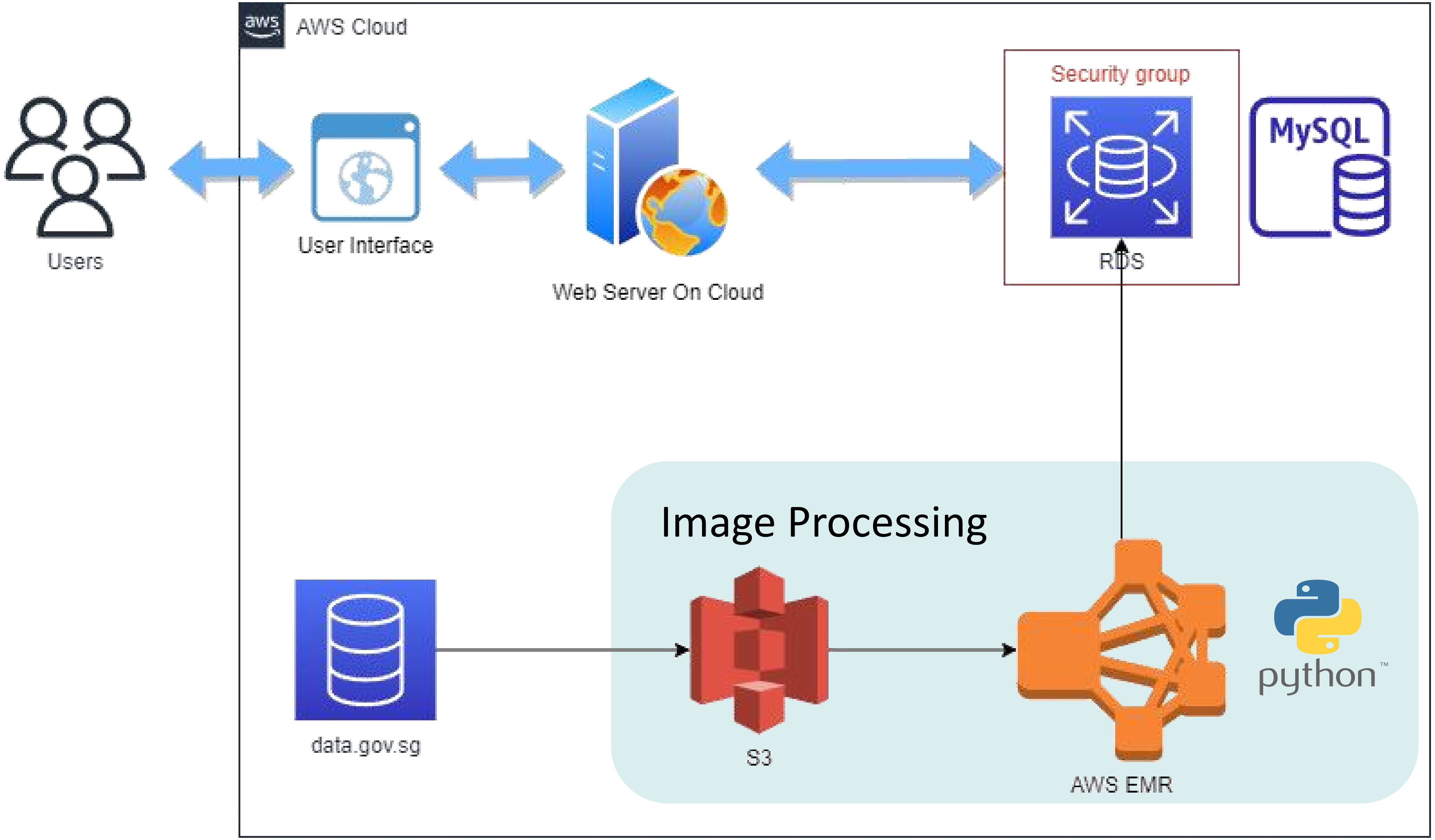


- NodeJS & Express
  - Serve static web content
  - Process HTTP requests from frontend
  - Use MySQL statement to get data from RDS

IaaS

Cloud Architecture

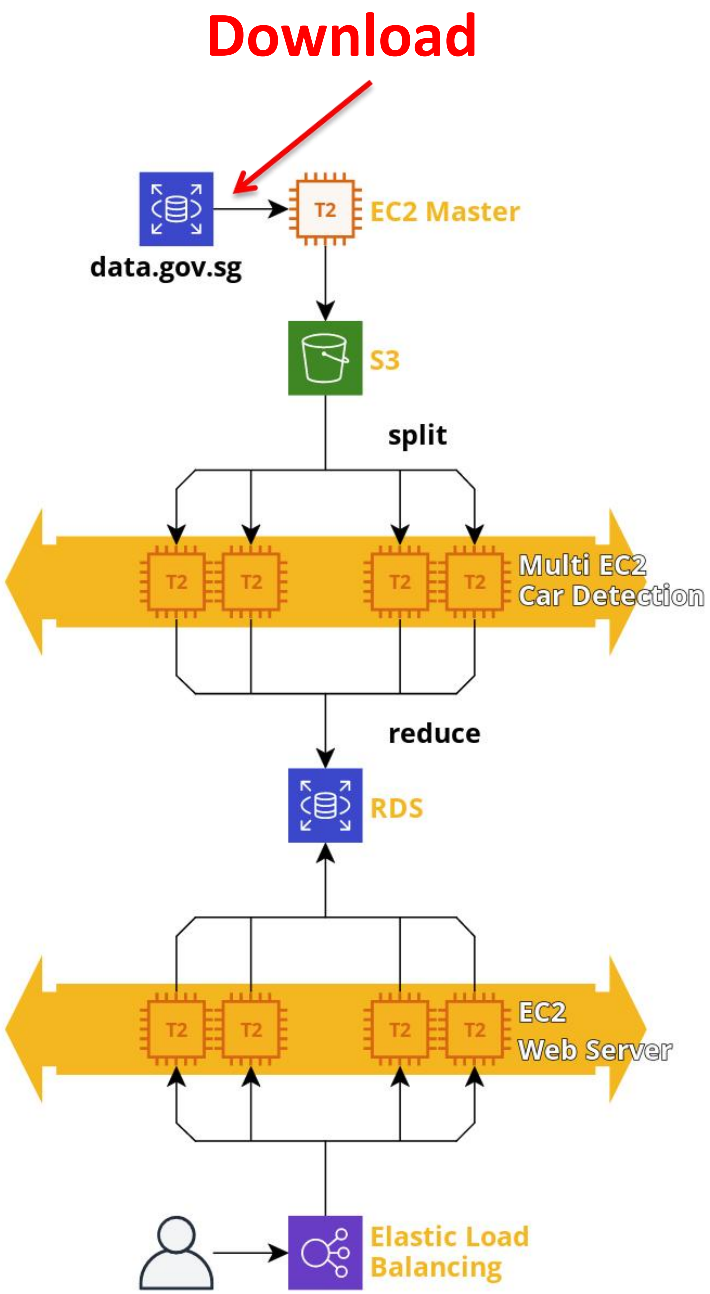
- **Image Processing On Cloud**



- Python
  - OpenCV API
- Map&Reduce
  - Large Dataset
  - Parallel makes Efficiences

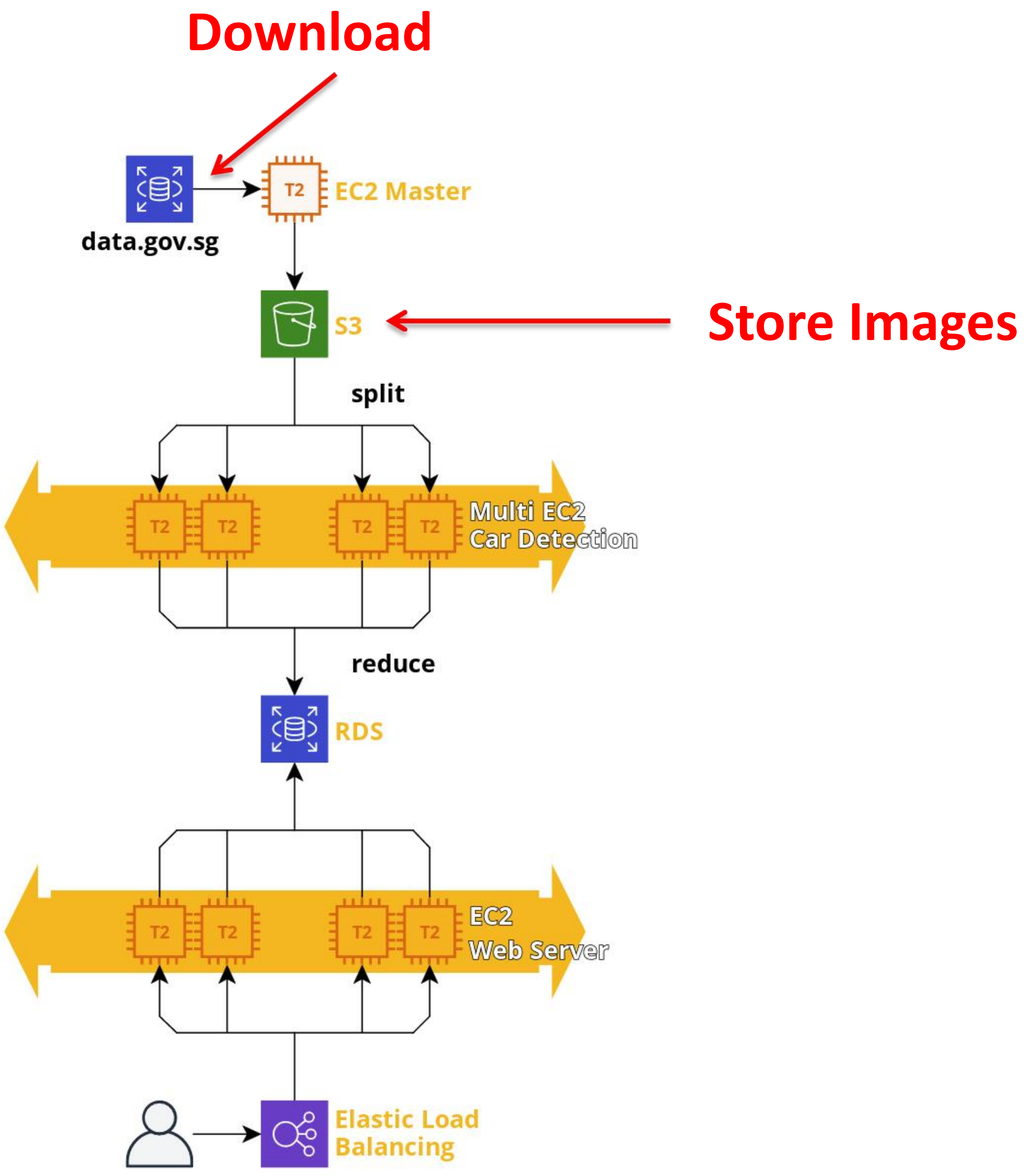
Cloud Architecture

- **Workflow**



Cloud Architecture

- **Workflow**

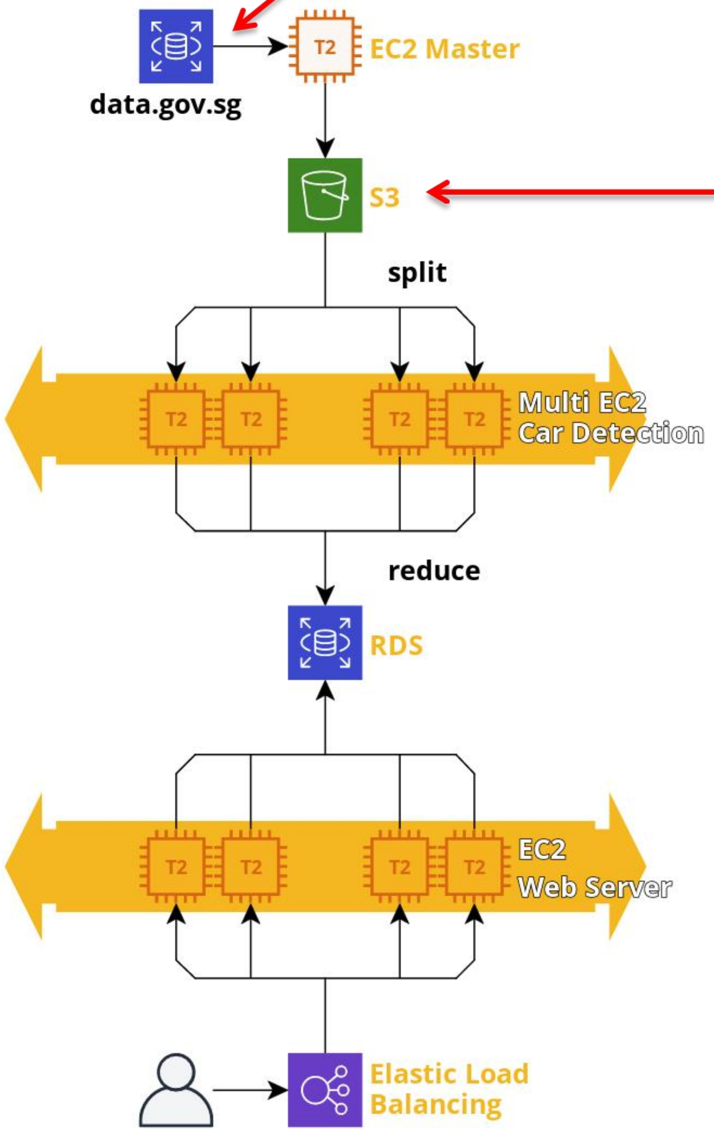


Download

Store Images

Cloud Architecture

- Workflow



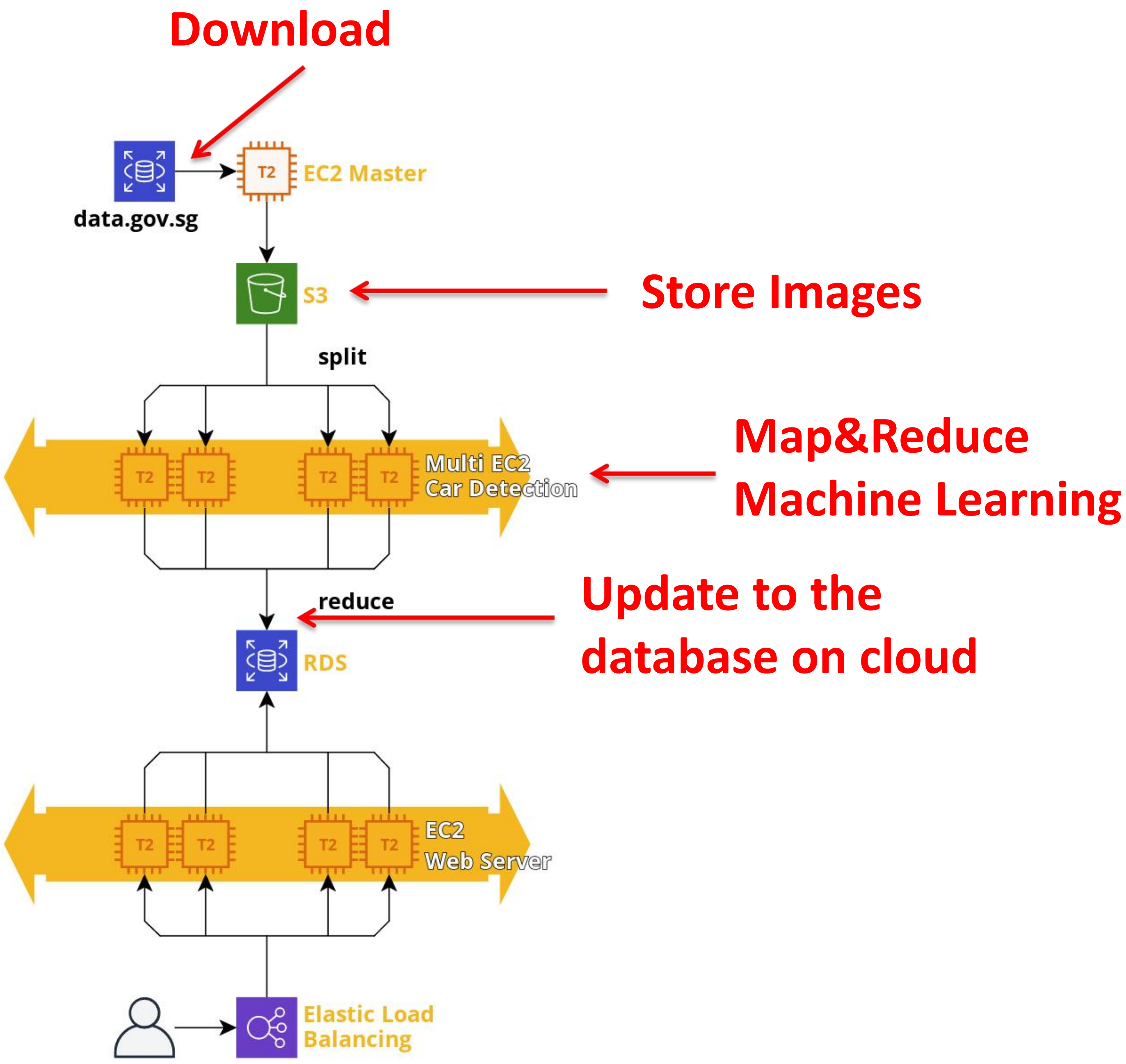
Download

Store Images

Map&Reduce
Machine Learning

Cloud Architecture

- Workflow



Download

Store Images

Map&Reduce
Machine Learning

Update to the
database on cloud

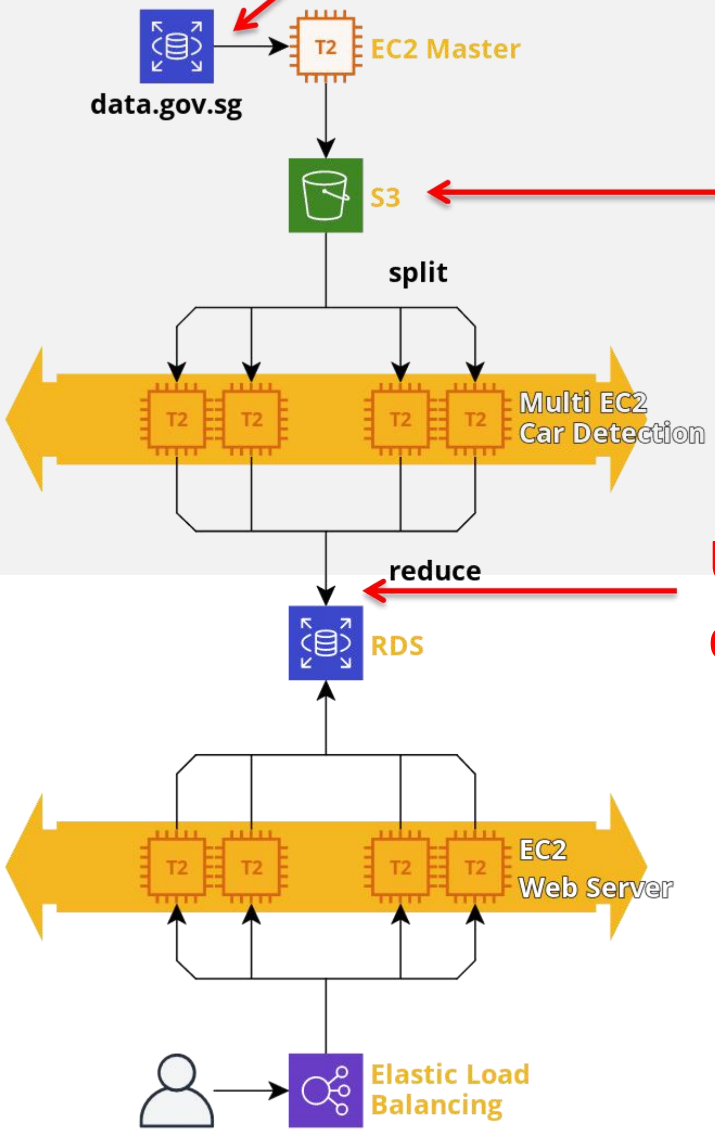
Cloud Architecture

- Workflow

Keep Running in the background

Download and process every 5 mins

Download



Store Images

Map&Reduce Machine Learning

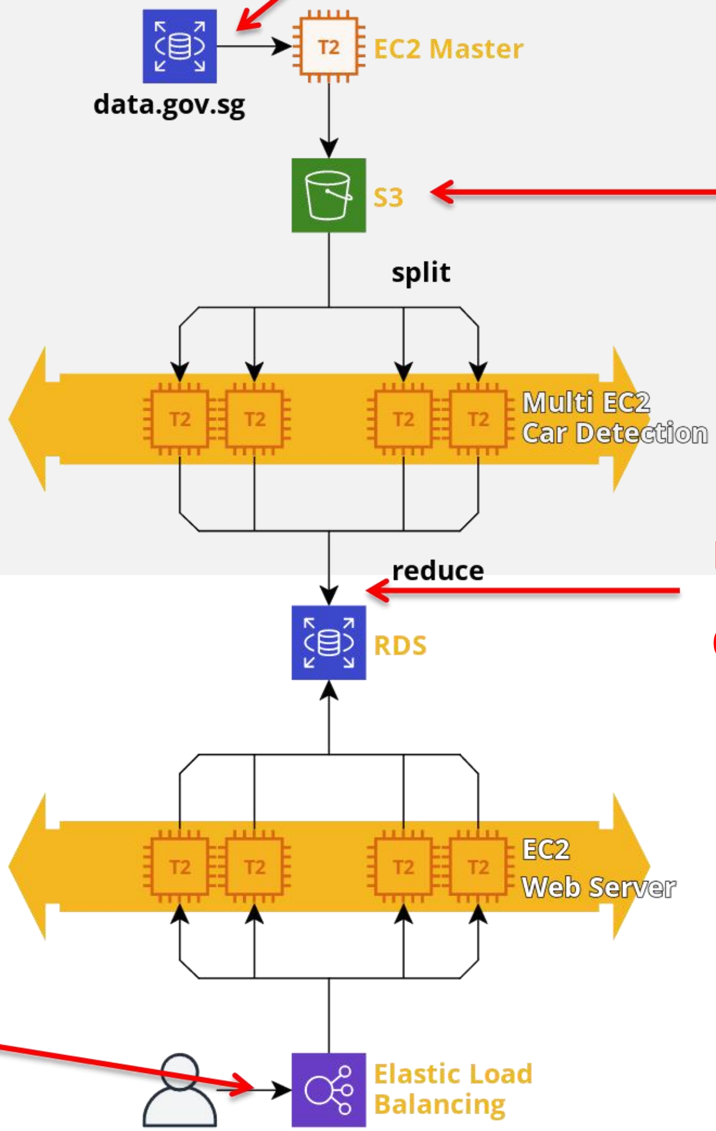
Update to the database on cloud

Cloud Architecture

- Workflow

Keep Running in the background

Download and process every 5 mins



Download

Store Images

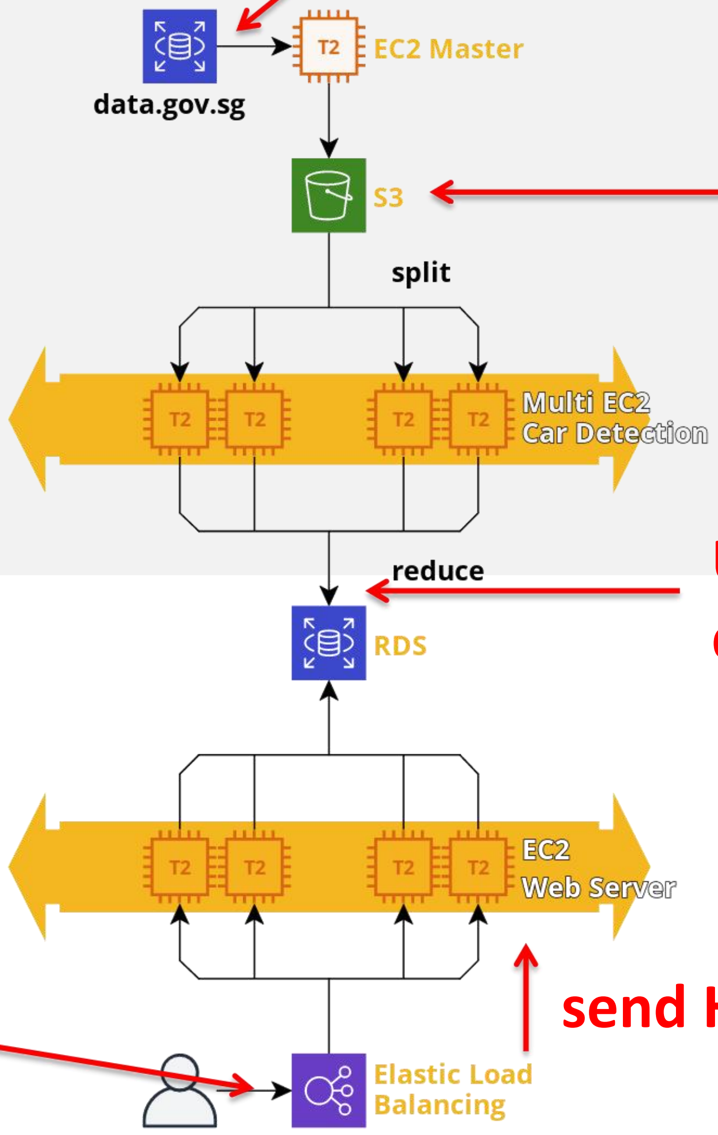
Map&Reduce Machine Learning

Update to the database on cloud

Users interact with Web UI

Cloud Architecture

- Workflow



Download

Keep Running in the background

Download and process every 5 mins

Store Images

Map&Reduce
Machine Learning

Update to the database on cloud

send HTTP request

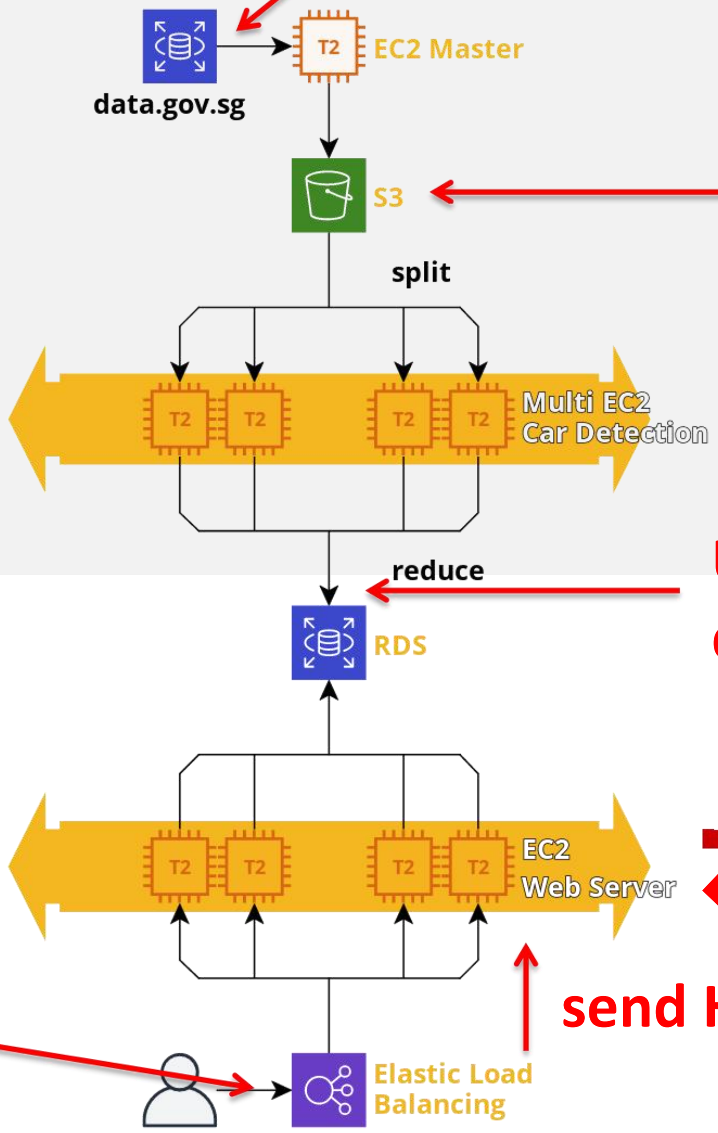
Users interact with Web UI

Cloud Architecture

- Workflow

Keep Running in the background

Download and process every 5 mins



Download

Store Images

Map&Reduce Machine Learning

Update to the database on cloud

parse HTTP request

send HTTP request

Users interact with Web UI

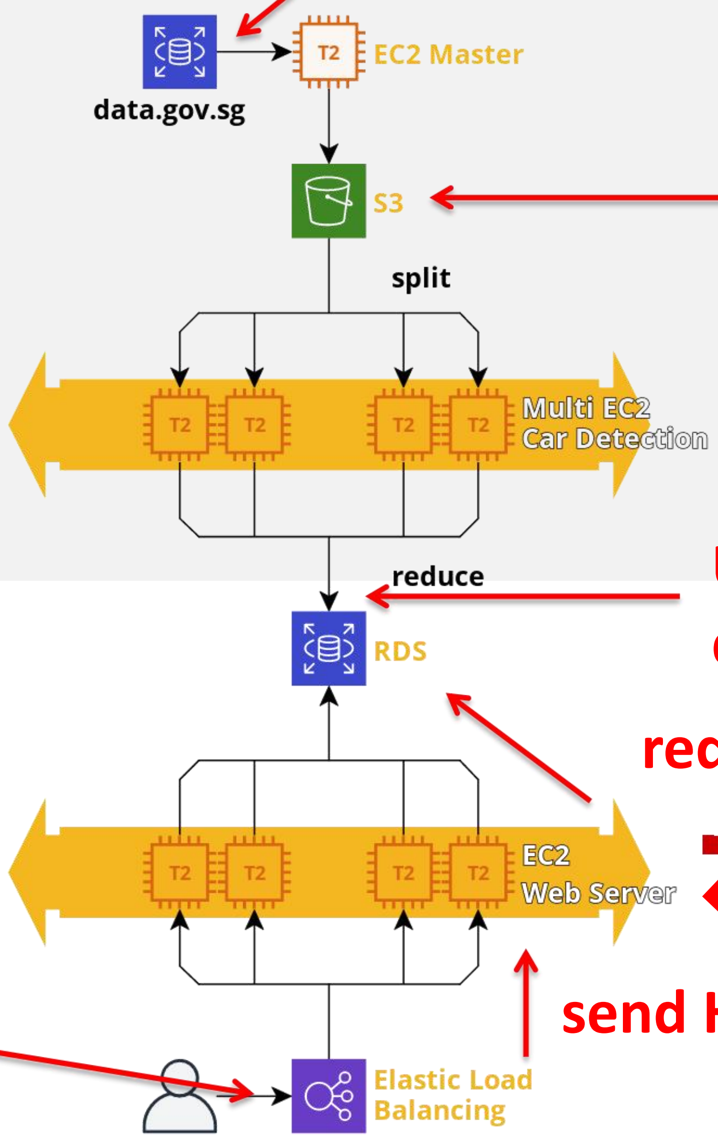
Cloud Architecture

- Workflow

Download

Keep Running in the background

Download and process every 5 mins



Store Images

Map&Reduce
Machine Learning

Update to the database on cloud

request traffic info(MySQL)

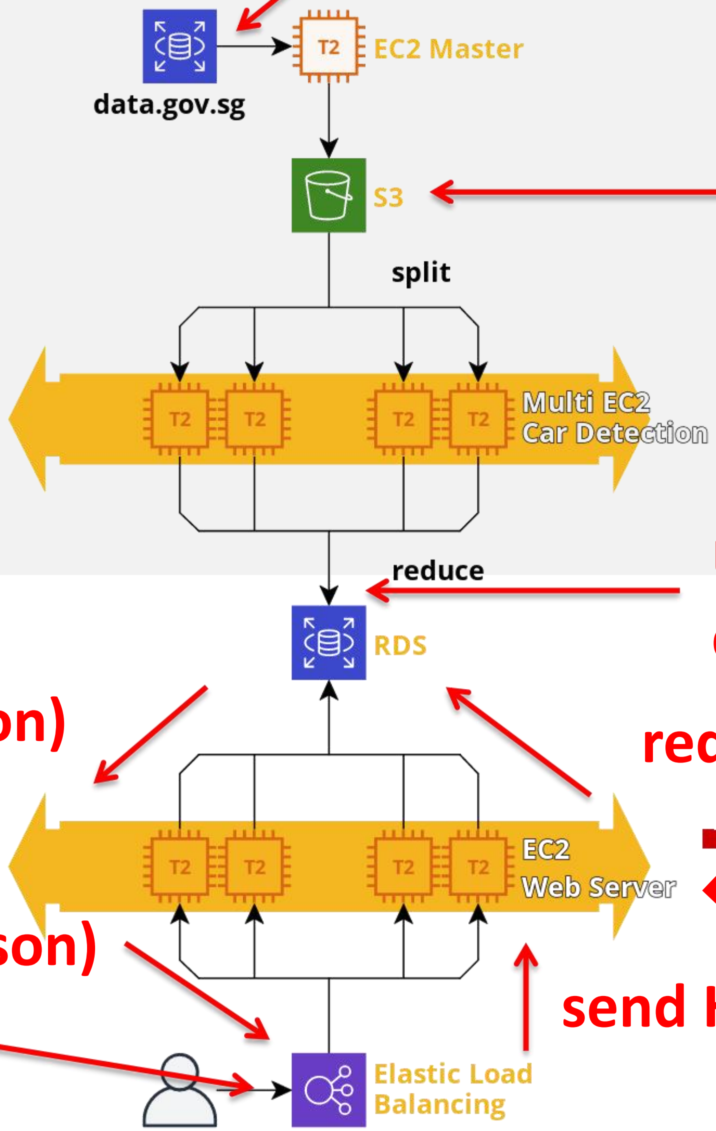
parse HTTP request

send HTTP request

Users interact with Web UI

Cloud Architecture

- Workflow



Download

Keep Running in the background

Download and process every 5 mins

Store Images

Map&Reduce Machine Learning

Update to the database on cloud

return traffic info(json)

request traffic info(MySQL)

parse HTTP request

return traffic info(json)

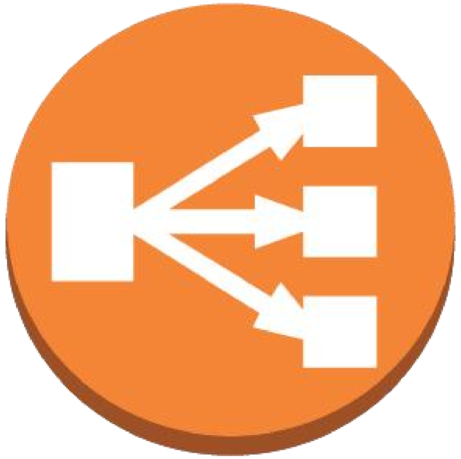
send HTTP request

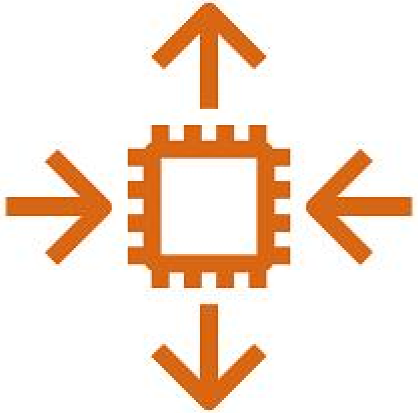
Users interact with Web UI

Cloud Architecture

- **Elasticity & Scalability**

- Web Server
  - Elastic Load Blancer & Auto Scalling Group
    - Automaticlly Scales capacity based on the incoming traffic(Scalability)
    - Distrubute incoming traffic to mutiple resources(Elasticity)





Cloud Architecture

- **Elasticity & Scalability**

- Image Processing
  - Use pyspark for parallelism
  - AWS EMR
    - Automatically scaling the compute resources based on the load(Elasticity)
    - various big data processing frameworks(Scalability)

Outline

- Overview
- UI Design Diagram
- Image Processing
  - Car Detection
  - SQL
- Cloud Architecture
  - Web Server On Cloud
  - Image Processing On Cloud
  - Elasticity & Scalability
- Implementation Plan

DEMO

Implementation Plan

- **Frontend----Zhu Yundian，Liu Chenghang**

- Finished work
  - The website’s basic framework
  - Displaying the map, route and icons

- Unfinished work and problems
  - Connecting to the database
  - Website beautification

Implementation Plan

- **Image Processing----Yao Chenxuan**

- Finished work
  - Successfully identifying the car number in every intersection

- Problems
  - Can not find a reliable method to know the orientation of the camera

Implementation Plan

- **Cloud Architecture----Tang Tang**

- Finished work
  - Building a simple cloud architecture

- Unfinished work
  - Adding more details and functions.

Thank You!