



THE GOOD, THE BAD AND THE UGLY

---

ELASTICSEARCH



# BASICS



# WHAT IS ELASTICSEARCH?

Elasticsearch is a highly scalable open-source full-text search and analytics engine.

It allows you to store, search, and analyze big volumes of data quickly and in near real time.



elasticsearch

JSON  
query

```
$> curl -XPOST -d ...
```

Elasticsearch

HTTP API  
(JAVA API)

JSON  
response

```
{
  "query" : {
    "bool" : {
      "must" : [ {
        "geo_bbox" : {
          "geo" : {
            "top_left" : [ 48.8438105, 2.2635173 ],
            "bottom_right" : [ 48.8883955, 2.3857733 ]
          }
        }
      }, {
        "terms" : {
          "id" : [ "1", "4", "7" ]
        }
      }
    ]
  },
  "aggregations" : {
    "geo" : {
      "geohash_grid" : {
        "precision" : 7
      }
    }
  }
}
```



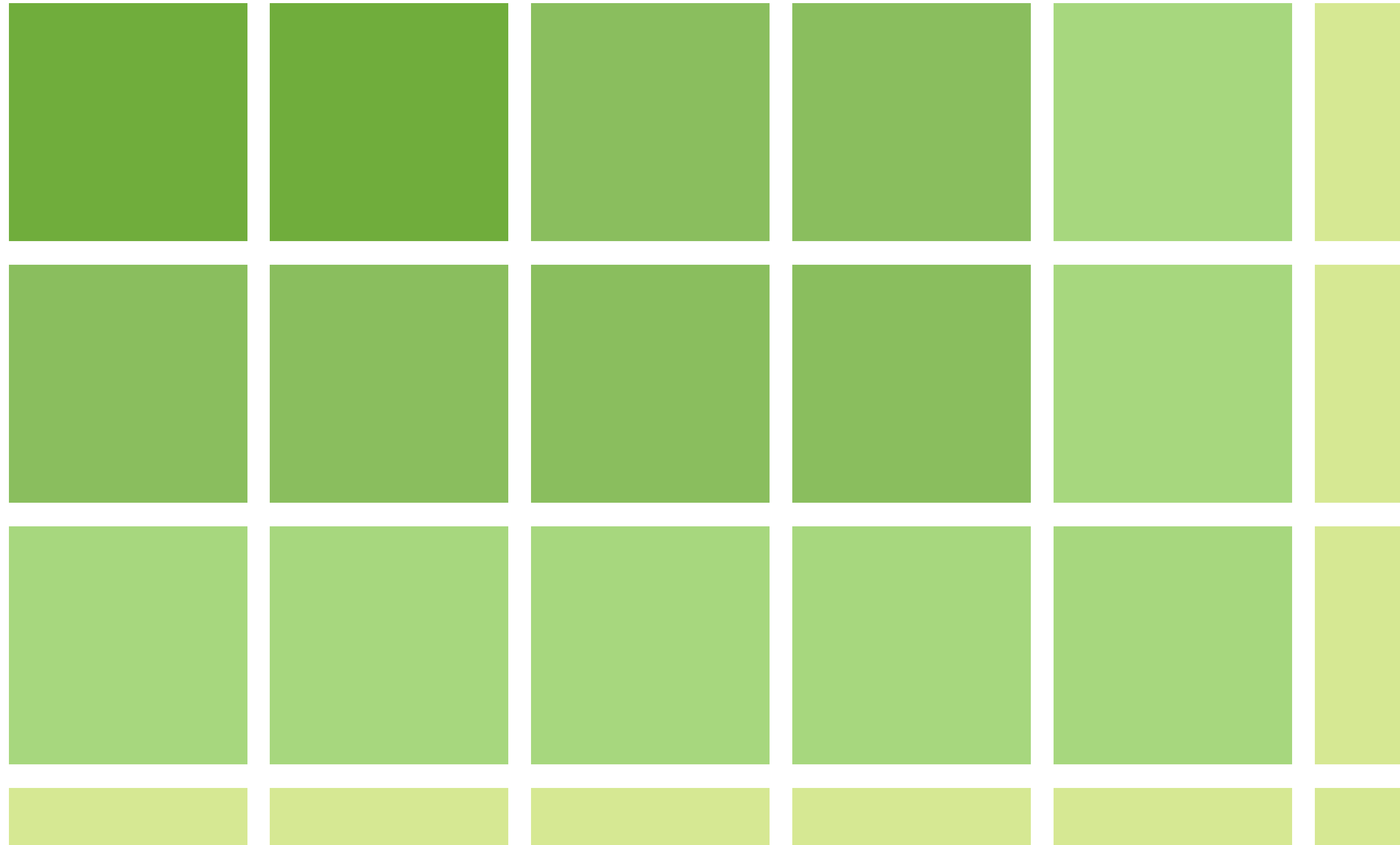


# CLUSTER

**A COLLECTION OF NODES**

Arranged around a "Master"  
Require quorum

Internode communications  
"Discovery"





NODE

**A SINGLE SERVER**



Black Widow  
Daredevil  
Doctor Strange  
Odin  
Groot  
John Constantine  
...



# INDEX

A COLLECTION OF DOCUMENTS



Index  
Search  
Update  
Delete



TYPE

A CATEGORY OF DOCUMENTS



Business partitions





# DOCUMENT

**A BASIC UNIT OF INFORMATION**



Flat hierarchy  
Inverted index

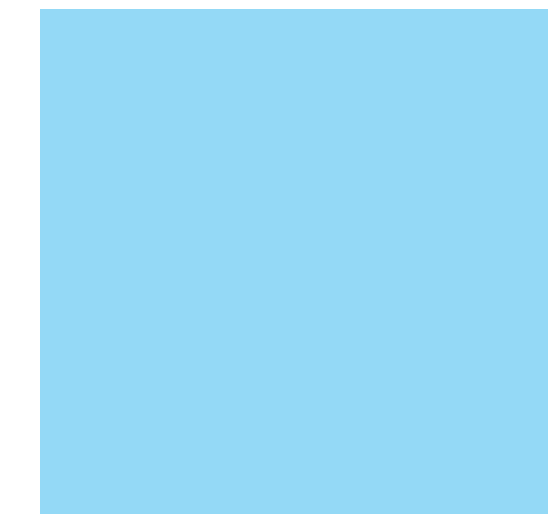
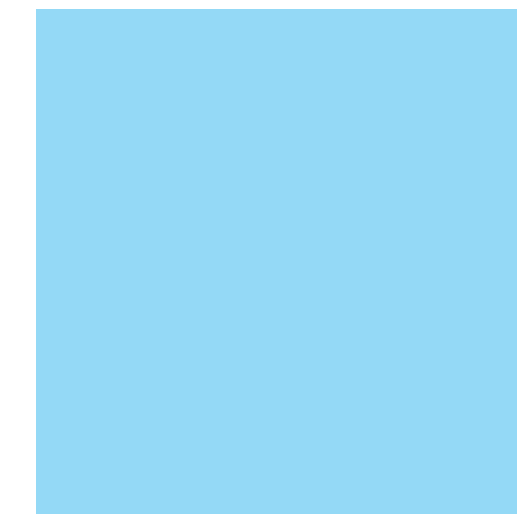
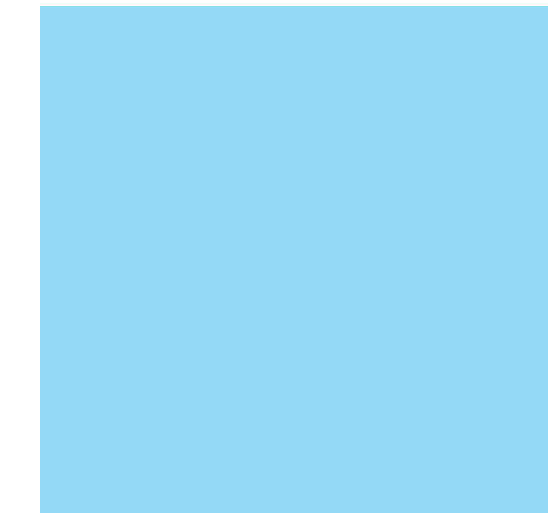
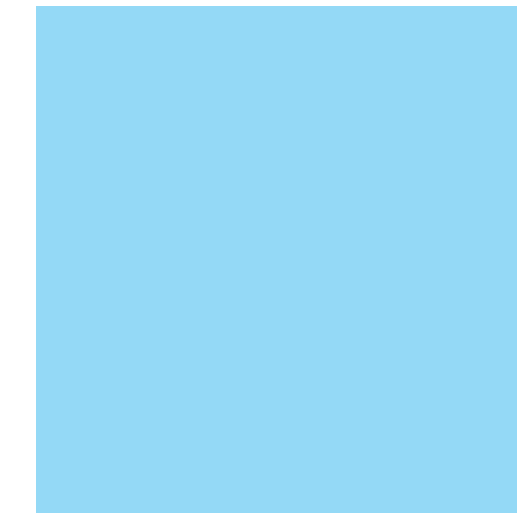
# SHARDS & REPLICAS

## PARTITIONS OF AN INDEX

Horizontal split/scale  
Parallelize operations



Primary Shards



Replica Shards



# INDEXING A DOCUMENT

```
curl -XPUT localhost:9200/my_index/my_type/my_id -d '{body}'
```

```
PUT /my_index/my_type/my_id
```

```
{  
  "name": "Tuco",  
  "realName": "Eli Wallach"  
}
```

SEARCH

**BLACK = CURL**



**SCALA = WHITE**



# SEARCHING FOR NON-NULL FIELDS

```
search in vesselIndex / "vessel" query {  
  existsQuery("lastPosition")  
} > 2805 Hits
```

**EXISTS QUERY**



elasticsearch

# SEARCHING FOR NON-NULL FIELDS

```
{  
  "query" : {  
    "exists" : {  
      "field" : "lastPosition"  
    }  
  }  
}
```

**EXISTS QUERY**



elasticsearch

# SEARCHING FOR AN EXACT MATCH

```
search in vesselIndex / "vessel" query {  
  termQuery("name", "naroon")  
} > 1 Hit
```

```
search in vesselIndex / "vessel" query {  
  termQuery("flagName", "iran")  
} > 14 Hits
```

**TERM QUERY**



elasticsearch

# SEARCHING FOR AN EXACT MATCH

```
{  
  "query" : {  
    "term" : {  
      "name" : "naroon"  
    }  
  }  
}
```

**TERMQUERY**





elasticsearch

# SELECTING RANGES

```
search in portCallIndex / "port_call" query {  
  rangeQuery("start").gte("now-3d/d")  
} > 752 Hits
```

**RANGE QUERY**



elasticsearch

# SELECTING RANGES

```
{  
  "query" : {  
    "range" : {  
      "start" : {  
        "from" : "now-3d/d",  
        "to" : null,  
        "include_lower" : true,  
        "include_upper" : true  
      }  
    }  
  }  
}
```

**RANGE QUERY**



elasticsearch

# SEARCHING FOR APPROXIMATE MATCH

```
search in vesselIndex / "vessel" query {  
  fuzzyQuery("name", "narroon").fuzziness(Fuzziness.ONE)  
} > 1 Hit
```

```
search in portCallIndex / "port_call" query {  
  fuzzyQuery("start", "2016-01-01").fuzziness(Fuzziness.build("2d"))  
} > 739 Hits
```

**FUZZY QUERY**



elasticsearch

# SEARCHING FOR APPROXIMATE MATCH

```
{  
  "query" : {  
    "fuzzy" : {  
      "name" : {  
        "value" : "narroon",  
        "fuzziness" : "1"  
      }  
    }  
  }  
}
```

FUZZY QUERY

```
{  
  "query" : {  
    "fuzzy" : {  
      "start" : {  
        "value" : "2016-01-01",  
        "fuzziness" : "2d"  
      }  
    }  
  }  
}
```



elasticsearch

# COMBINING QUERIES

```
search in vesselIndex / "vessel" query {  
  bool {  
    must(  
      termQuery("lastPosition.currentCargo.loaded", true),  
      rangeQuery("lastPosition.speed").lte(1.5)  
    ) should(  
      termQuery("lastPosition.currentCargo.commodity", "other"),  
      termQuery("beta", true)  
    )  
  } minimumShouldMatch(1)  
} > 102 Hits
```

**BOOL QUERY**





elasticsearch

# COMBINING QUERIES

```
{
  "query" : {
    "bool" : {
      "must" : [ {
        "term" : {
          "lastPosition.currentCargo.loaded" : true
        }
      }, {
        "range" : {
          "lastPosition.speed" : {
            "from" : null,
            "to" : 1.5,
            "include_lower" : true,
            "include_upper" : true
          }
        }
      }
    ],
    "should" : [ {
      "term" : {
        "lastPosition.currentCargo.commodity" : "other"
      }
    }, {
      "term" : {
        "beta" : true
      }
    }
  ],
  "minimum_should_match" : "1"
}
}
```

BOOL QUERY

# AGGREGATIONS



elasticsearch

## BUCKET

Group documents by  
matching criterion

## METRIC

Compute metrics over  
a set of documents

## PIPELINE

Aggregate the output of  
other aggregations



elasticsearch

# COMPUTING AVERAGE

```
search in portCallIndex / "port_call" query {  
  rangeQuery("flowQuantity.volume").gte(0.0).includeLower(false)  
} aggregations {  
  avgAggregation("average-loading").field("flowQuantity.volume")  
} size(0)
```

> 128 160 m3 (806 126 bbl)

**AVG AGGREGATION**



elasticsearch

# COMPUTING AVERAGE

```
{
  "size" : 0,
  "query" : {
    "range" : {
      "flowQuantity.volume" : {
        "from" : 0.0,
        "to" : null,
        "include_lower" : false,
        "include_upper" : true
      }
    }
  },
  "aggregations" : {
    "average-loading" : {
      "avg" : {
        "field" : "flowQuantity.volume"
      }
    }
  }
}
```

**AVG AGGREGATION**





elasticsearch

# GROUPING AND SORTING

```
search in vesselIndex / "vessel" aggregations {  
  termAggregation("per-cargo").field("portCallInfo.lastPortCall.currentCargo.commodity") aggregations {  
    topHitsAggregation("first")  
      .fetchSource(includes = Array("name"), excludes = Array(""))  
      .sort(field.sort("cargoMetrics.capacity").order(SortOrder.DESC))  
      .size(1)  
    }  
  }  
} size(0)
```

> Overseas Laura Lynn

> Alex

> TA'KUNTAH

**TERM AGGREGATION**  
**TOP HIT AGGREGATION**



elasticsearch

# GROUPING AND SORTING

```
{
  "size" : 0,
  "aggregations" : {
    "per-cargo" : {
      "terms" : {
        "field" : "portCallInfo.lastPortCall.currentCargo.commodity"
      },
      "aggregations" : {
        "first" : {
          "top_hits" : {
            "size" : 1,
            "_source" : {
              "includes" : [ "name" ],
              "excludes" : [ "" ]
            },
            "sort" : [ {
              "cargoMetrics.capacity" : {
                "order" : "desc"
              }
            ]
          }
        }
      }
    }
  }
}
```

TERM AGGREGATION  
TOP HIT AGGREGATION



elasticsearch

# GEOGRAPHIC CLUSTERING

```
search in positionIndex / "position" query {  
  rangeQuery("speed").lte(0.3)  
} aggregations {  
  // precision(5) == 5km x 5km cells  
  geoHashGridAggregation("grid").field("geo").precision(5) aggregations {  
    geoCentroidAggregation("cluster").field("geo")  
  }  
} size(0)
```

**GEO HASH GRID AGGREGATION**  
**GEO CENTROID AGGREGATION**



elasticsearch

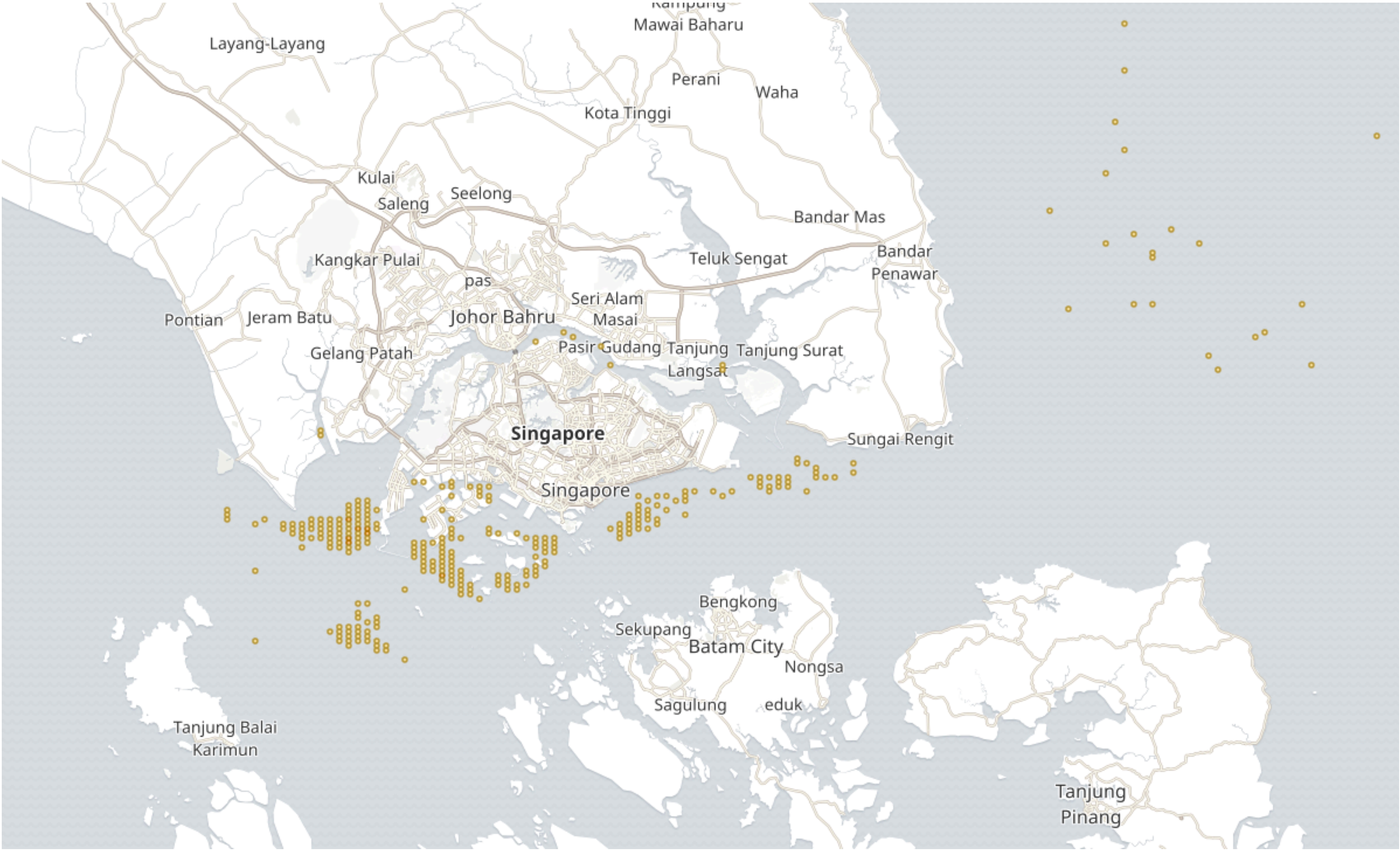
# GEOGRAPHIC CLUSTERING

```
{
  "size" : 0,
  "query" : {
    "range" : {
      "speed" : {
        "from" : null, "to" : 0.3,
        "include_lower" : true,
        "include_upper" : true
      }
    }
  },
  "aggregations" : {
    "grid" : {
      "geohash_grid" : {
        "field" : "geo"
      },
      "aggregations" : {
        "cluster" : {
          "geo_centroid" : {
            "field" : "geo"
          }
        }
      }
    }
  }
}
```

**GEO HASH GRID AGGREGATION**  
**GEO CENTROID AGGREGATION**



elasticsearch



# CONCLUSION



elasticsearch

# QUESTIONS?



elasticsearch

# DOCUMENTATION

**ELASTICSEARCH** <https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html>

**SCALA DSL** <https://github.com/sksamuel/elastic4s>

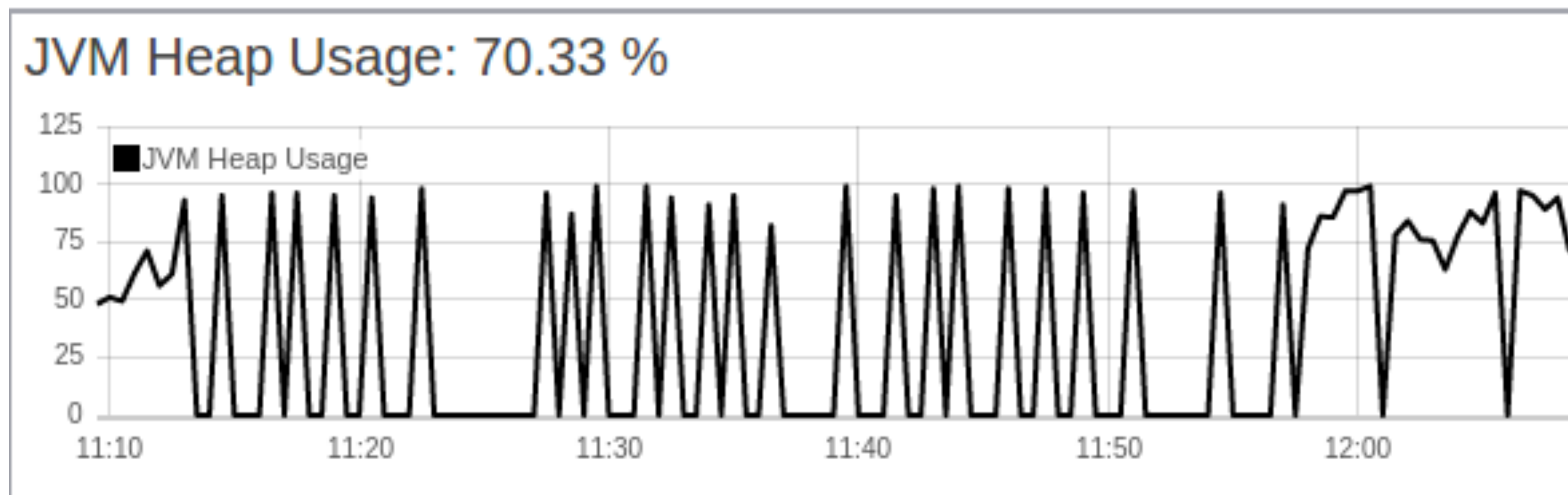
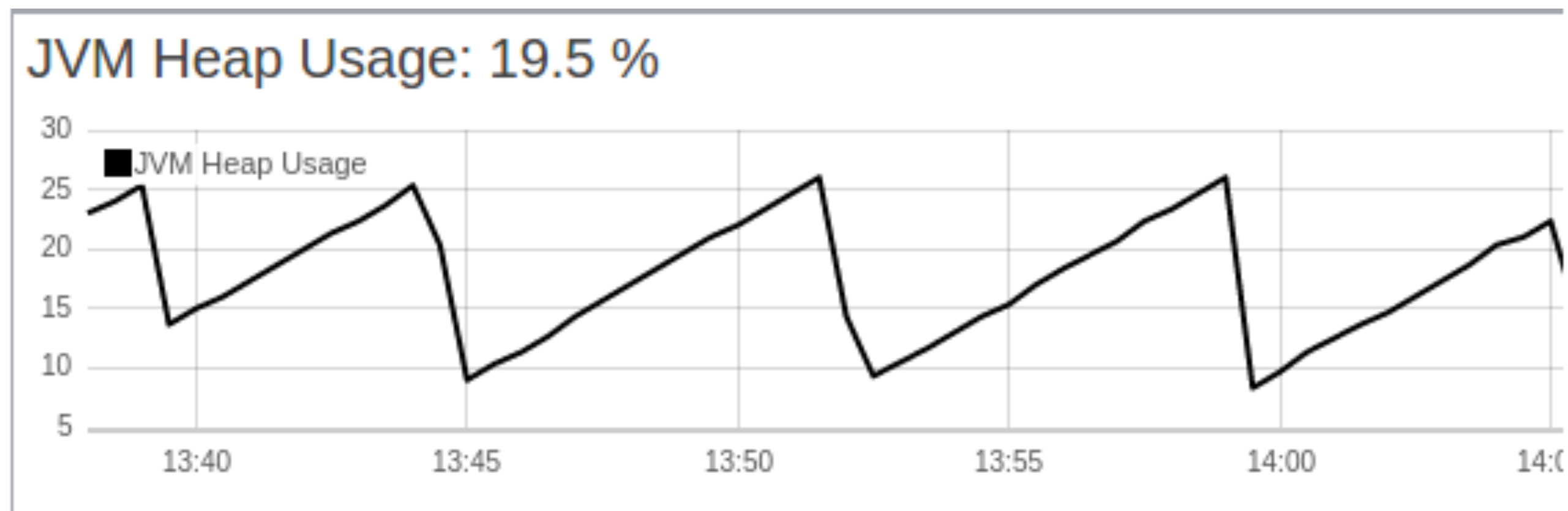
**SLIDES AND CODE** <https://github.com/fiahil/Talks>





elasticsearch

# NOTES ON COMMON ELASTICSEARCH PROBLEMS

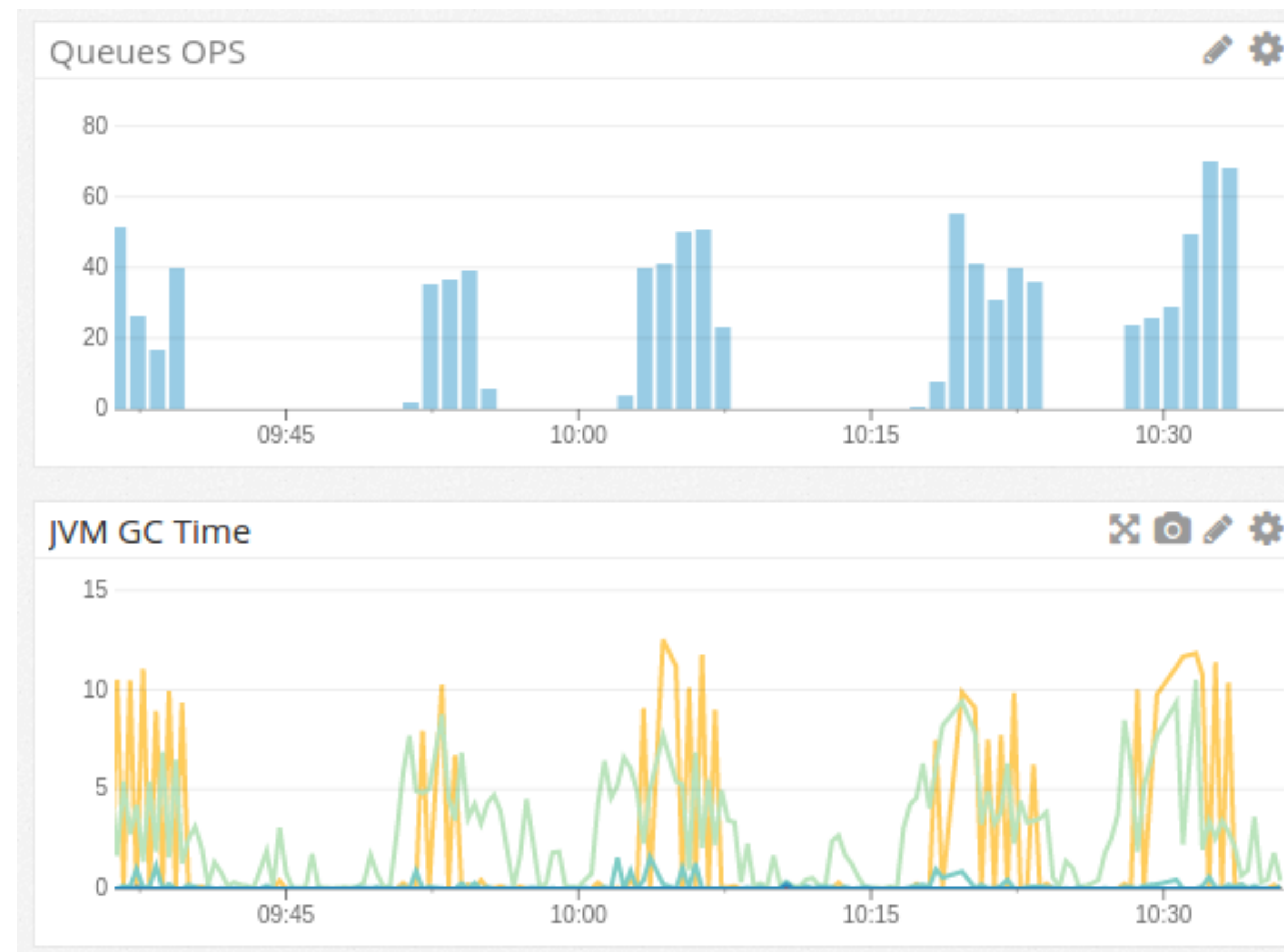


**PLEASE UPGRADE YOUR CLUSTER**



elasticsearch

# NOTES ON COMMON ELASTICSEARCH PROBLEMS



**PLEASE UPGRADE YOUR CLUSTER**



elasticsearch

# MAPPINGS

```
}
},
"4405": { + },
"4414": { + },
"4421": { + },
"4427": { + },
"4431": { + },
"4435": { + },
"4436": { + },
"4446": { + },
"4455": { + },
"4459": { + },
"4476": { + },
"4484": { + },
"4487": { + },
"4490": { -
  "properties": { -
    "zone": { -
      "properties": { -
        "name": { -
          "type": "string"
        },
        "fullname": { -
          "type": "string"
        },
        "id": { -
          "type": "long"
        },
        "resourceType": { -
          "type": "string"
        }
      }
    }
  },
  "id": { -
    "type": "long"
  }
}
},
"4492": { -
  "properties": { -
    "zone": { -
      "properties": { -
        "name": { -
          "type": "string"
        },
        "fullname": { -
          "type": "string"
        },
        "id": { -
          "type": "long"
        },
        "resourceType": { -
          "type": "string"
        }
      }
    }
  }
}
```

( NOT REALLY AN ISSUE )