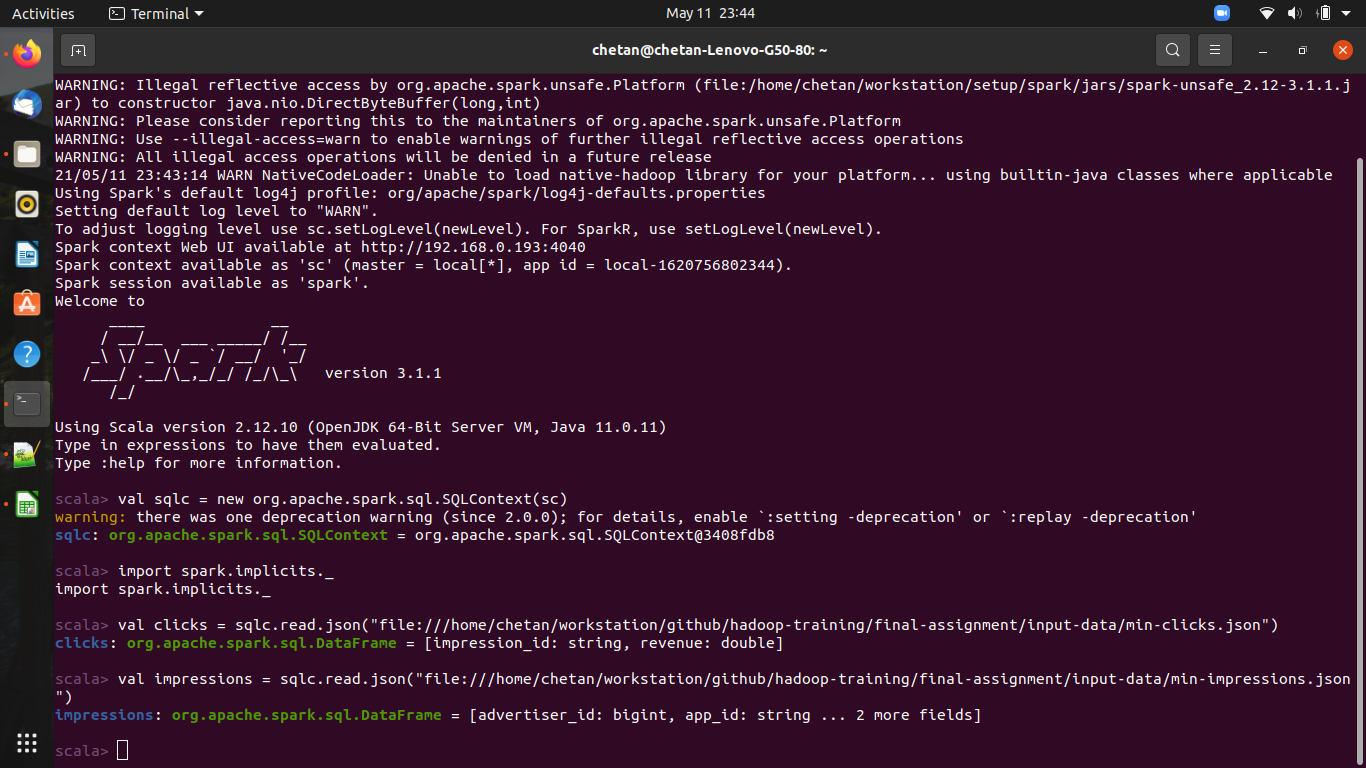
**1. Read events stored in JSON files**

**commands run -**

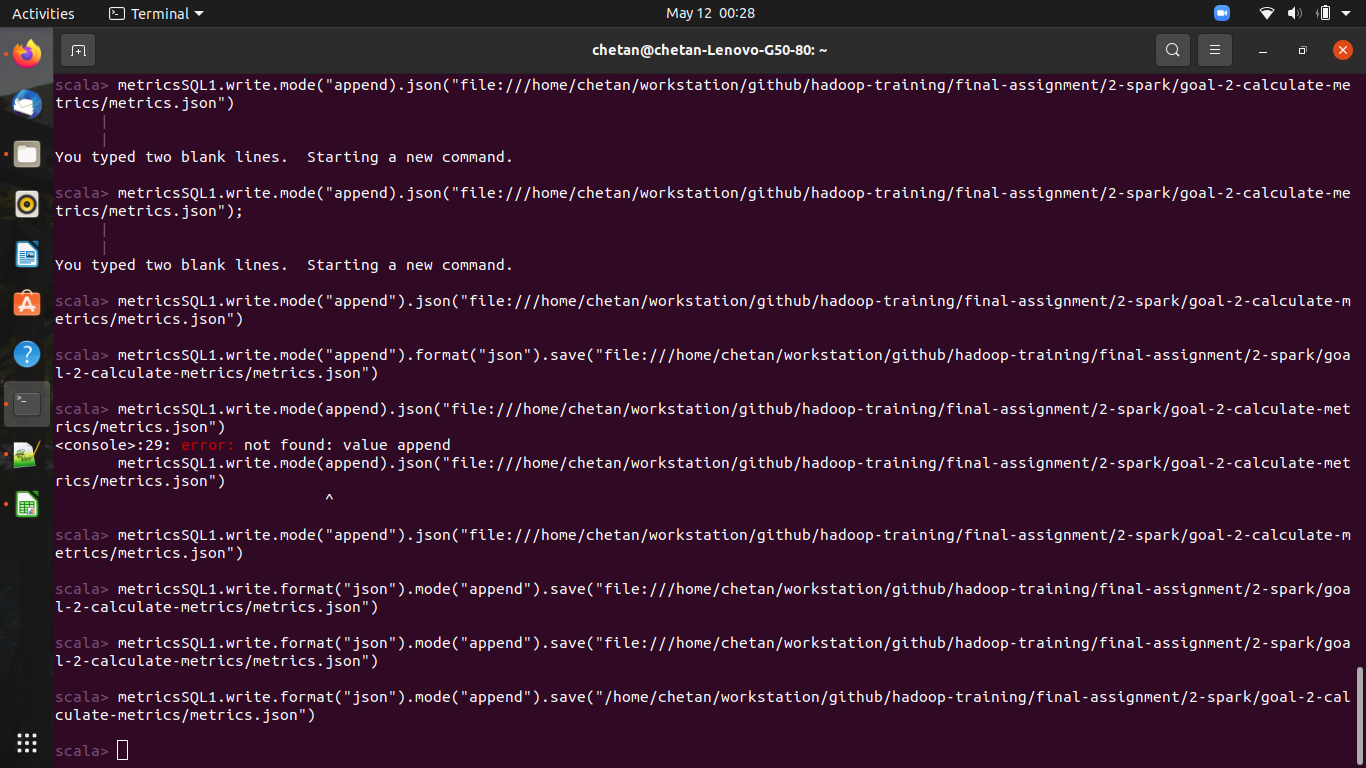
* val sqlc = new org.apache.spark.sql.SQLContext(sc)
* import spark.implicits.\_
* val clicks = sqlc.read.json("[file:///home/chetan/workstation/github/hadoop-training/final-assignment/input-data/min-clicks.json](../input-data/min-clicks.json)")
* val impressions = sqlc.read.json("file:///home/chetan/workstation/github/hadoop-training/final-assignment/input-data/min-impressions.json")



### 2. Calculate metrics for some dimensions

**commands run -**

* clicks.registerTempTable("temp\_clicks")
* impressions.registerTempTable("temp\_impressions")
* val metricsSQL = sqlc.sql("select m.app\_id, m.country\_code, count(m.id) as impression, count (c.impression\_id) as clicks, sum (c.revenue) from temp\_impressions m left join temp\_clicks c on m.id = c.impression\_id group by m.app\_id, m.country\_code")
* metricsSQL.write.format("json").mode("append").save("/home/chetan/workstation/github/hadoop-training/final-assignment/2-spark/goal-2-calculate-metrics/metrics.json")



### **3. Make a recommendation for the top 5 advertiser\_ids to display for each app and country combination.**

**commands run -**

* val recommendationSQL = sqlc.sql("select m.app\_id, m.country\_code, m.advertiser\_id, sum (c.revenue) as t\_revenue from temp\_impressions m left join temp\_clicks c on m.id = c.impression\_id group by m.app\_id, m.country\_code, m.advertiser\_id order by m.app\_id, m.country\_code , t\_revenue desc")
* recommendationSQL.write.format("json").mode("append").save("/home/chetan/workstation/github/hadoop-training/final-assignment/2-spark/goal-3-top-recommendation")

