DE-3 Final Project

Ersan Kucukoglu

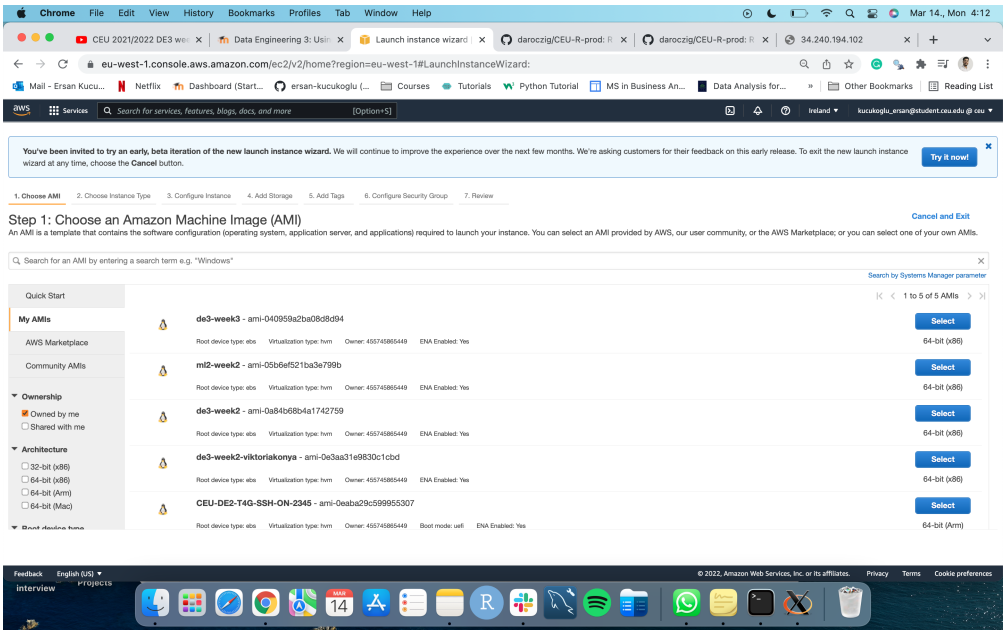
2022-03-26

- Instance ID: i-0338bc99ab45686e4

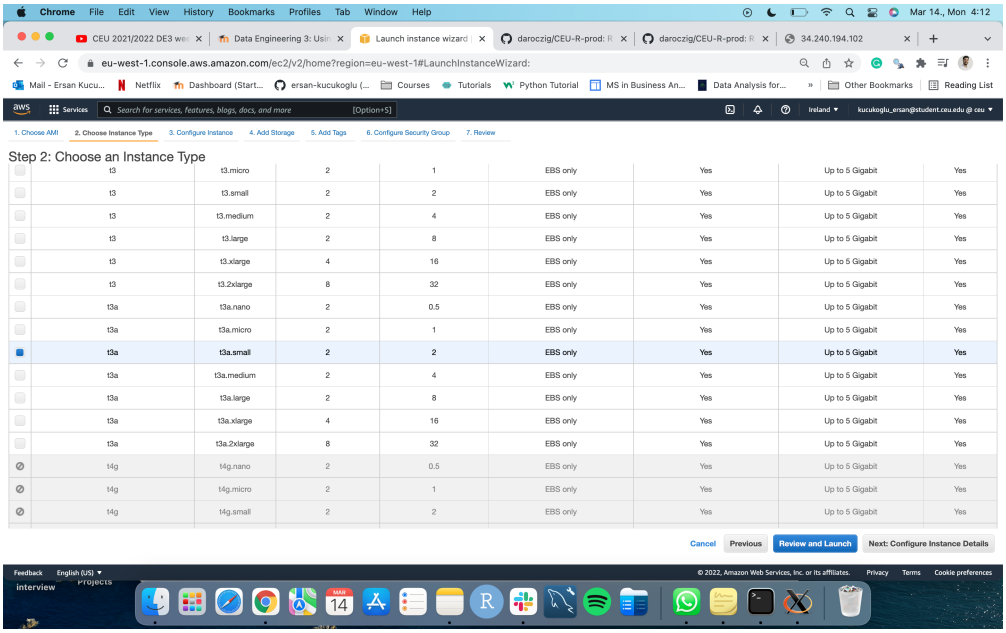
Introduction

The goal of this assignment is how to build data pipelines using Amazon Web Services and R, creating a stream processing application using the AWR.Kinesis R package's daemon + Redis.

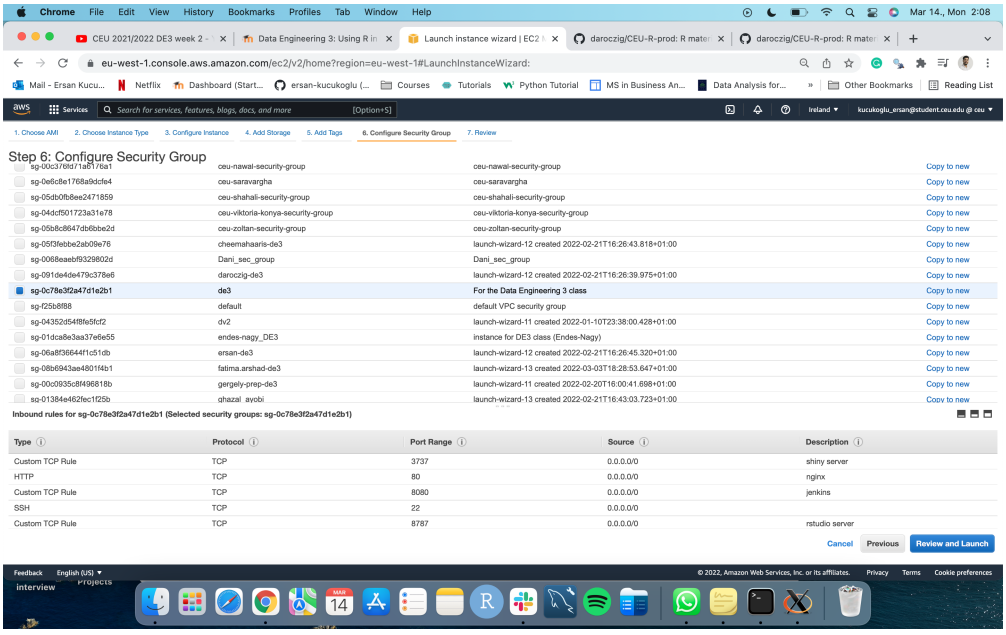
1. As a first step, I created EC2 node.To do that I selected de3-week3 AMI (already configured).



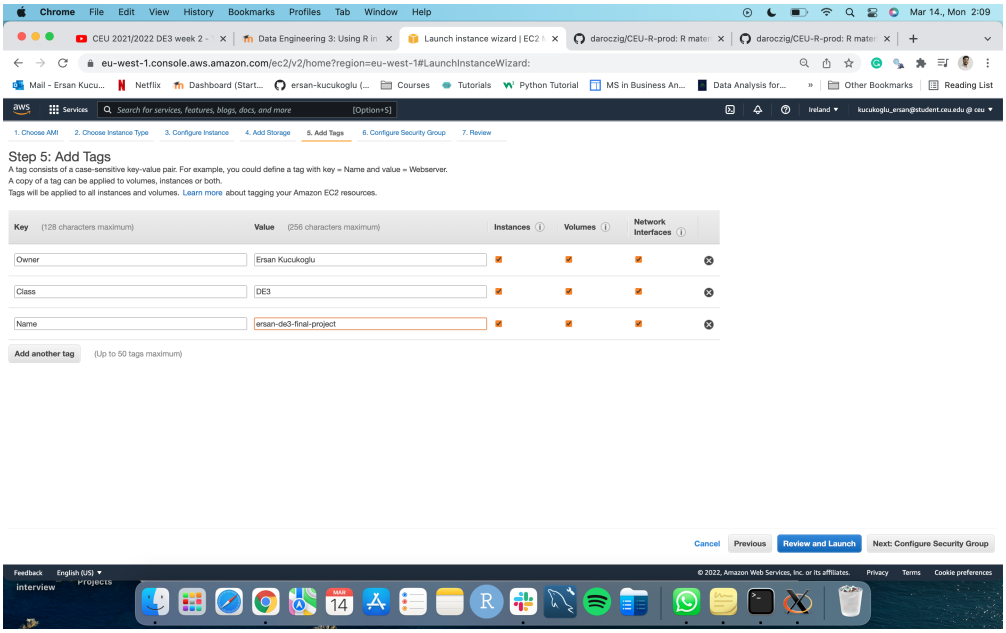
2. Then I selected an Instance Type t3a.small.



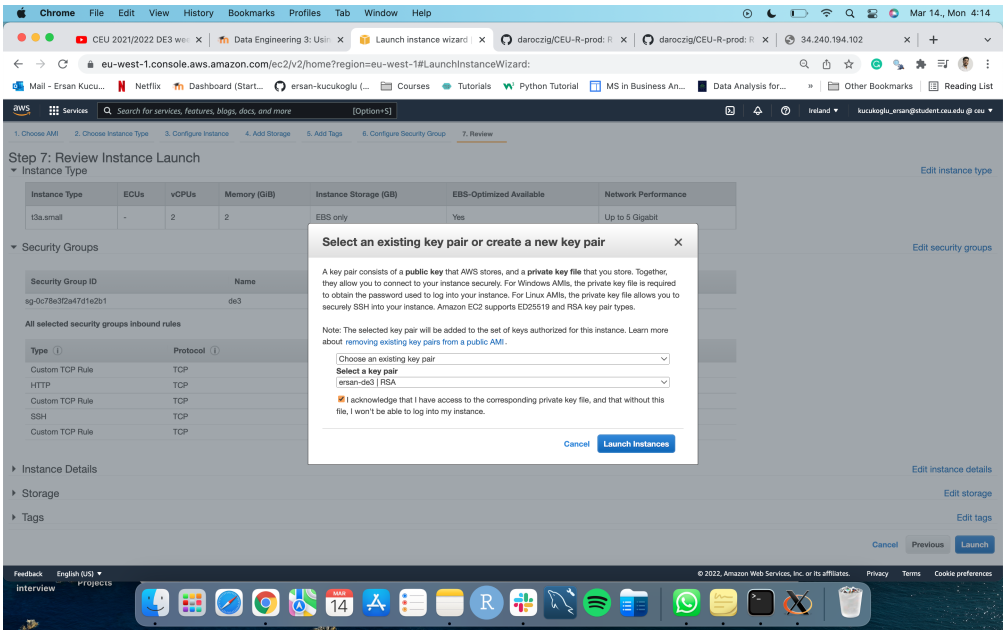
3. I selected the de3 security group which is shown below.



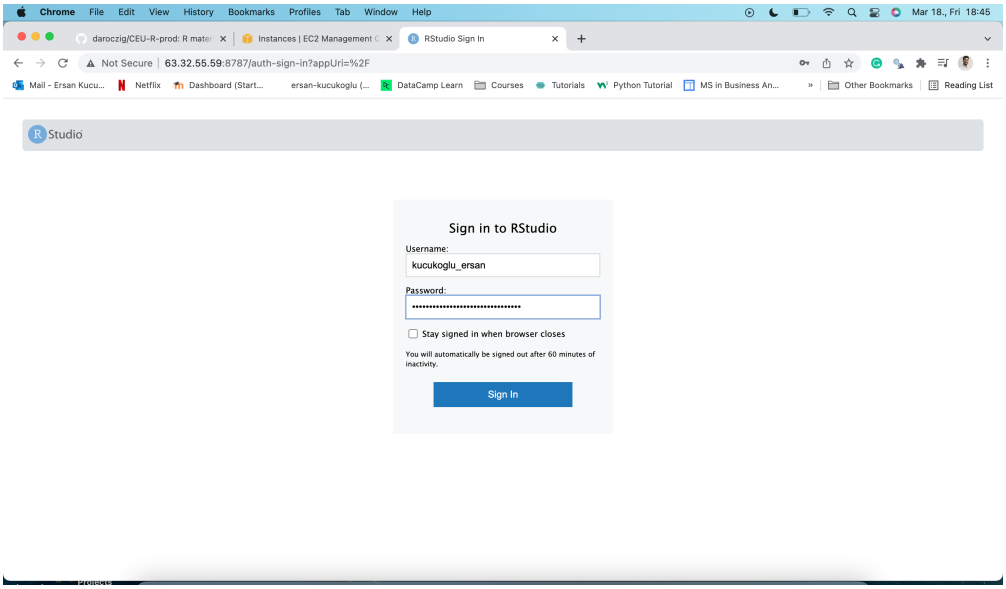
4. I added tags; owner, class, name.



5. I selected my existing key pair (ersan-de3 | RSA)



6. I opened rstudio server with my username and password



Stream Processor Daemon

7. After creating a folder (ersan-streamer), I created app.properties file in the folder.

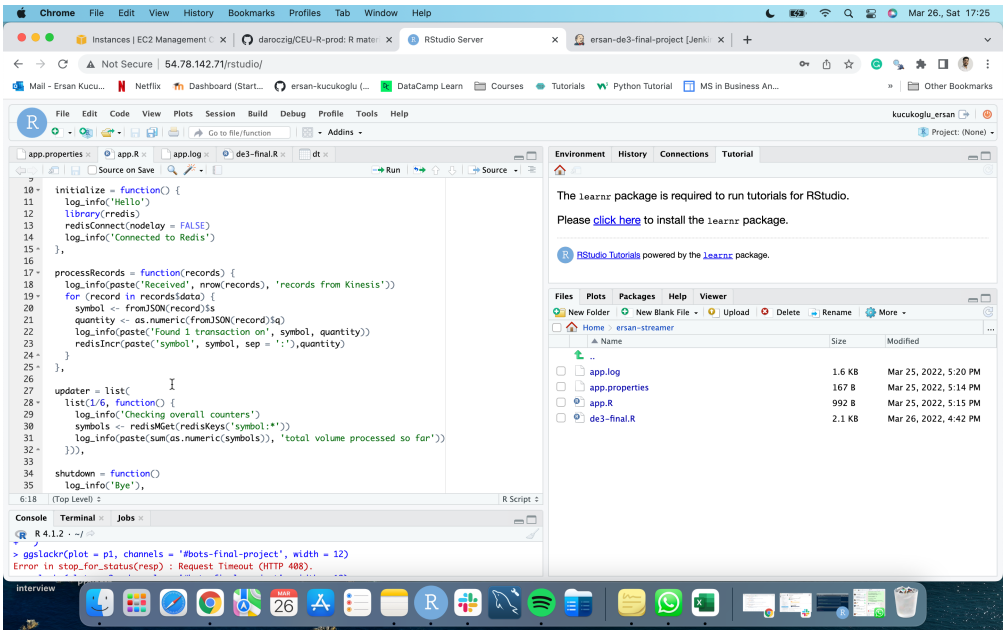
```
executableName = ./app.R
regionName = eu-west-1
streamName = crypto
applicationName = my_streamer_app_ersan
AWSCredentialsProvider = DefaultAWSCredentialsProviderChain
```

8. After that I created app.R file in the streamer folder.

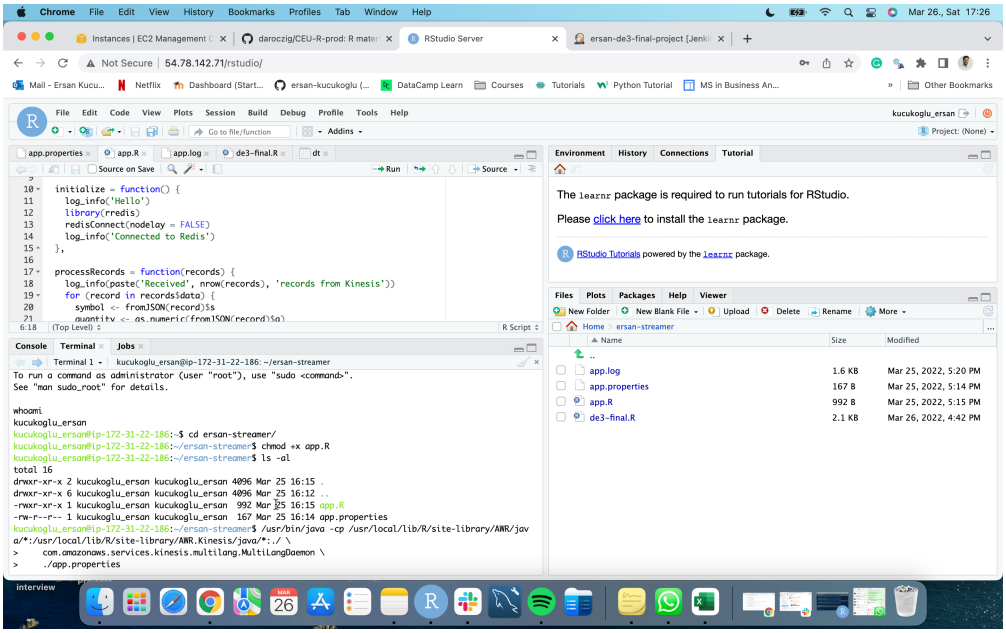
```
##app.R
#!/usr/bin/Rscript
library(logger)
log_appender(appender_file('app.log'))
library(AWR.Kinesis)
library(methods)
library(jsonlite)

kinesis_consumer(

  initialize = function() {
    log_info('Hello')
    library(rredis)
    redisConnect(nodelay = FALSE)
    log_info('Connected to Redis')
  },

  processRecords = function(records) {
    log_info(paste('Received', nrow(records), 'records from Kinesis'))
    for (record in records$data) {
      symbol <- fromJSON(record)$s
```

```
    quantity <- as.numeric(fromJSON(record)$q)
    log_info(paste('Found 1 transaction on', symbol, quantity))
    redisIncr(paste('symbol', symbol, sep = ':'),quantity)
  }
},

updater = list(
  list(1/6, function() {
    log_info('Checking overall counters')
    symbols <- redisMGet(redisKeys('symbol:*'))
    log_info(paste(sum(as.numeric(symbols)), 'total volume processed so far'))
  })),

shutdown = function()
  log_info('Bye'),

checkpointing = 1,
logfile = 'app.log')
```

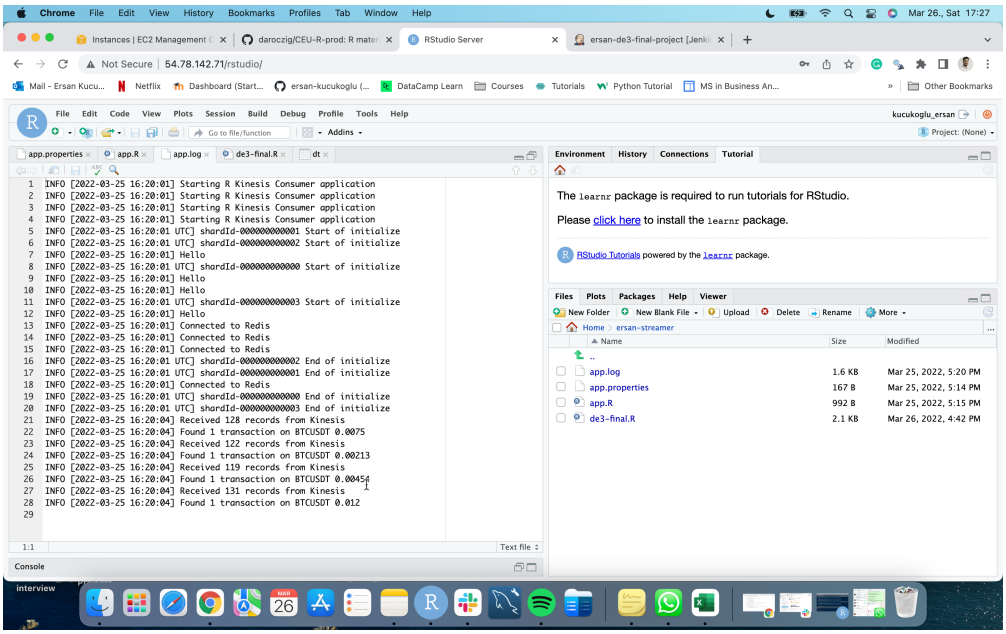
- See the files that I created in the ersan-streamer folder.



9. I ran the app using the Terminal.



10. I got the app.log file and got the crypto transactions data.

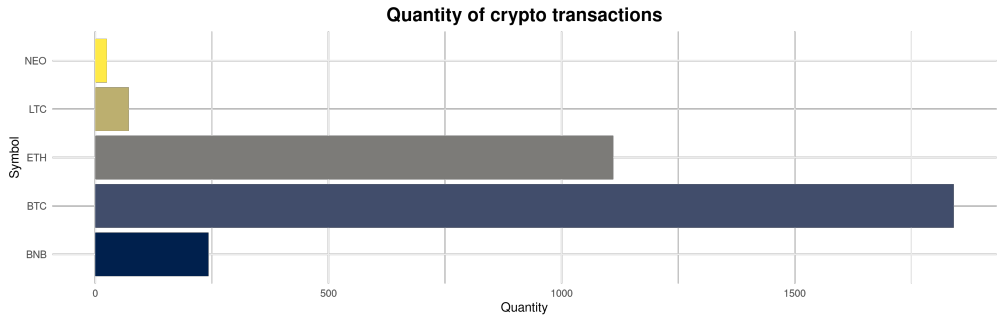


11. I used the streaming app to process the data from the Binance transactions and update the Redis values. I got the the total price of the crypto transactions, and the plots (Coins traded by value, Quantity of crypto transactions)

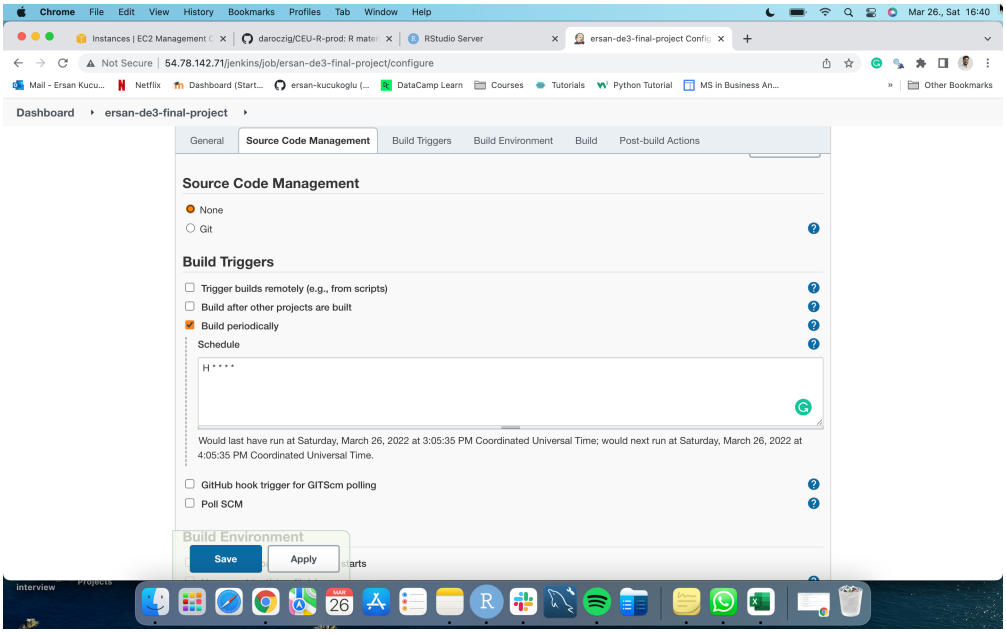
```
##de3-final.R
library(binancer)
library(jsonlite)
library(ggplot2)
library(data.table)
```

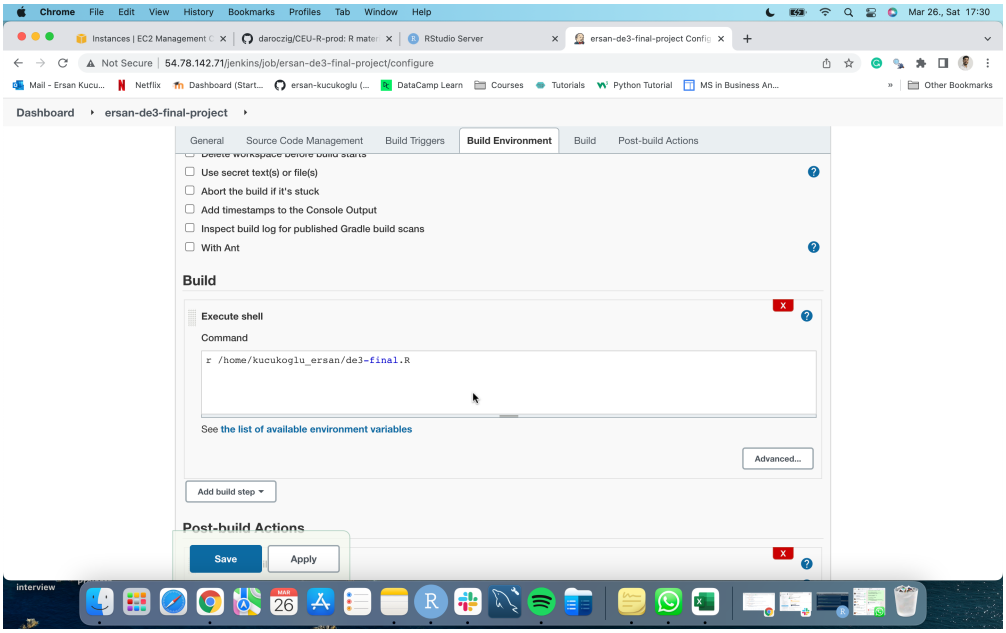
```
library(slackr)
library(scales)

library(botor)
botor(region = 'eu-west-1')


library(rredis)
redisConnect()

symbols <- redisMGet(redisKeys('symbol:*'))
symbols <- data.table(
  symbol = sub('^symbol:', '', names(symbols)),
  N = as.numeric(symbols))

symbols[, symbol := substr(symbol, 1, 3)]

symbols[, .(quantity = sum(N)), by = symbol]

prices <- binance_coins_prices()
dt <- merge(symbols, prices, by.x = 'symbol', by.y = 'symbol', all.x = TRUE, all.y = FALSE)

dt[, value := as.numeric(N) * usd]
total <- dt[, sum(value)]
totalquantity <- dt[, sum(N)]

dt[, sum(value), by = symbol]

# Print the message
print(paste0('The total price of the crypto transactions',
             ' is ',scales::dollar(total)))

p1 <- ggplot(dt)  +
  aes(x = symbol, fill = symbol, weight = N) +
  geom_bar() +
  scale_fill_viridis_d(option = "cividis", direction = 1) +
  labs(
    x = "Symbol",
    y = "Quantity",
    title = "Quantity of crypto transactions"
  ) +
  coord_flip() +
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 16L,
                              face = "bold",
                              hjust = 0.5)
  )
p1
p2 <- ggplot(dt) +
  aes(x = symbol, fill = symbol, weight = usd) +
  geom_bar() +
```

```
  scale_fill_brewer(palette = "YlOrRd", direction = 1) +
  labs(
    x = "Symbol",
    y = "Value (USD)",
    title = "Coins traded by value"
  ) +
  coord_flip() +
  theme_classic() +
  theme(
    legend.position = "none",
    plot.title = element_text(size = 15L,
                              face = "bold",
                              hjust = 0.5)
  )
p2
token <- ssm_get_parameter('slack')
slackr_setup(username = 'ersan', token = token, icon_emoji = ':jenkins-rage:')
slackr_msg(text = paste0('The total price of the crypto transactions',
                         ' is ',scales::dollar(total)),
           channel = '#bots-final-project')

ggslackr(plot = p1, channels = '#bots-final-project', width = 12)
ggslackr(plot = p2, channels = '#bots-final-project', width = 12)
```

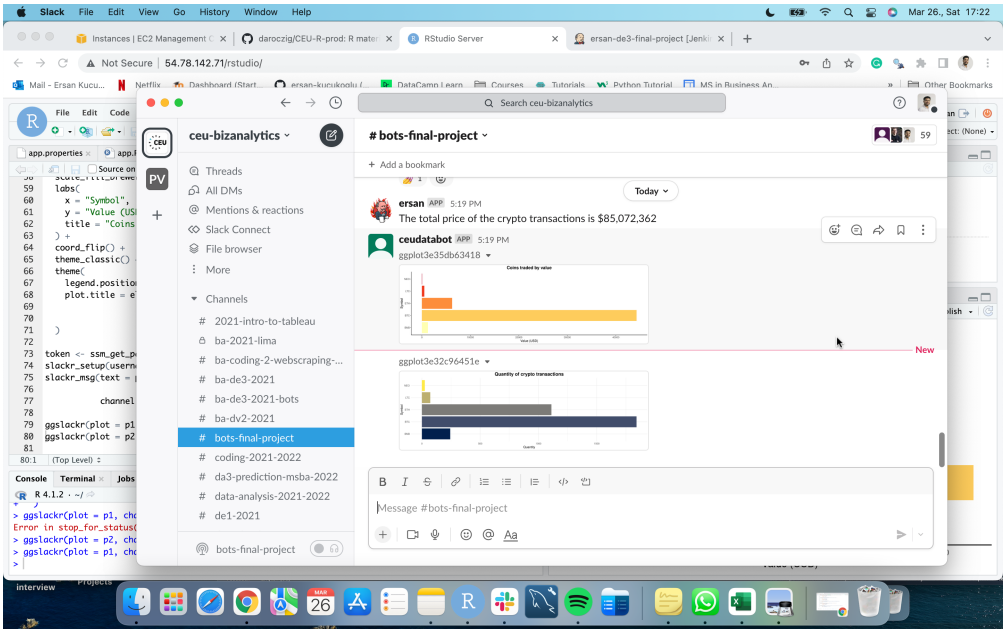


12. I logged in Jenkins, and created new item which is called ersan-de3-final-project. I created Jenkins job, scheduled hourly.





13. The total price of the transactions and plots are shared to the #bots-final-project channel.


Slack File Edit View Go History Window Help
Not Secure | 54.78.142.71/rstudio/
Search ceu-bizanalytics
ceu-bizanalytics
bots-final-project
Threads
All DMs
Mentions & reactions
Slack Connect
File browser
More
Channels
2021-intro-to-tableau
ba-2021-lima
ba-coding-2-webscraping-...
ba-de3-2021
ba-de3-2021-bots
ba-dv2-2021
bots-final-project
coding-2021-2022
da3-prediction-msba-2022
data-analysis-2021-2022
de1-2021
Add a bookmark
Today
ersan APP 5:19 PM
The total price of the crypto transactions is $85,072,362
ceudatabot APP 5:19 PM
ggplot3e35db63418
ggplot3e32c96451e
New
Message #bots-final-project