CEU - DE4 - Homework assignment

By Lisa Lang (1902224)

Instance_id: i-0090fe4e8582b9d3a

Stream processing application

7th April 2020

INSTRUCTIONS

Create a stream processing application using the AWR.Kinesis R package's daemon + Redis to record the overall amount of coins exchanged on Binance (per symbol) in the most recent micro-batch.
Create a Jenkins job that reads from this Redis cache and prints the overall value (in USD) of the transactions -- based on the coin prices reported by the Binance API at the time of request.
Create at least two more additional charts that display a metric you find meaningful, and report in the "#bots-final-project" Slack channel.
Make sure to clean-up your EC2 nodes, security groups, keys etc created in the past weeks.

GOALS

The goal of this assignment is to confirm that the students have a general understanding on how to build data pipelines using Amazon Web Services and R, and can actually implement a stream processing application (either running in almost real-time or batched/scheduled way) in practice.

DELIVERY METHOD

Create a PDF document that describes your solution and all the main steps involved with low level details: attach screenshots (including the URL nav bar and the date/time widget of your OS, so like full-screen and not area-picked screenshots) of your browser showing what you are doing in RStudio Server or eg Jenkins, make sure that the code you wrote is either visible on the screenshots, or included in the PDF. The minimal amount of screenshots are: EC2 creation, R code shown in your RStudio Server, Jenkins job config page, Jenkins job output, Slack channel notifications.

STOP the EC2 Instance you worked on, but don't terminate it, so we can start it and check how it works.

Submission Deadline: Midnight (CET) on April 19, 2020

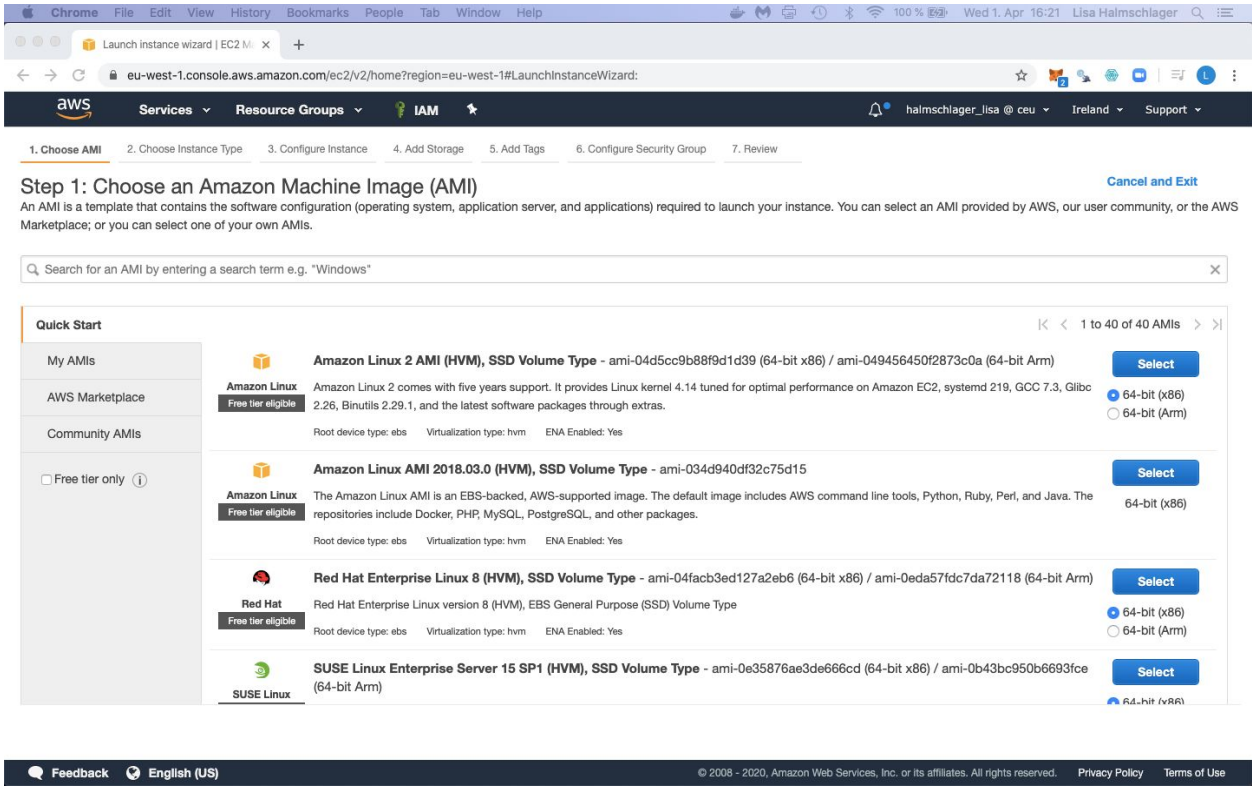
RESULTS

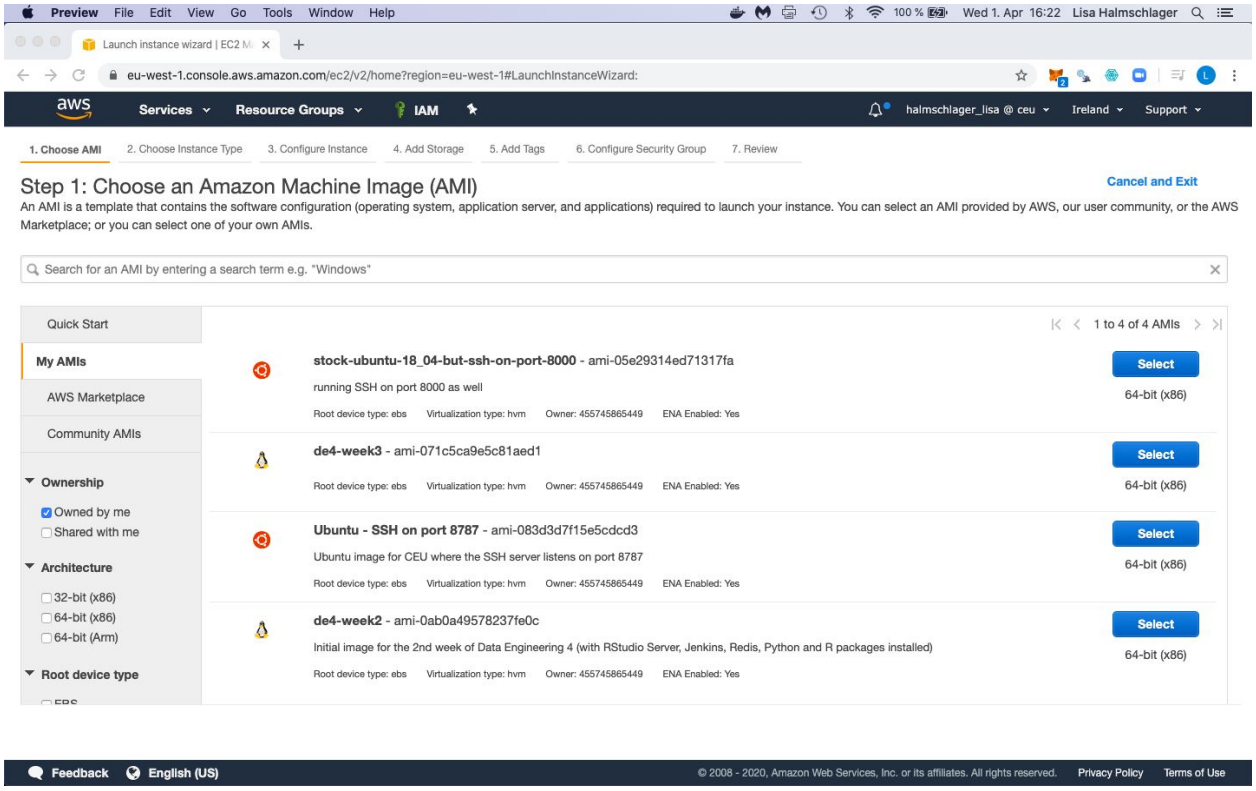
EC2 creation

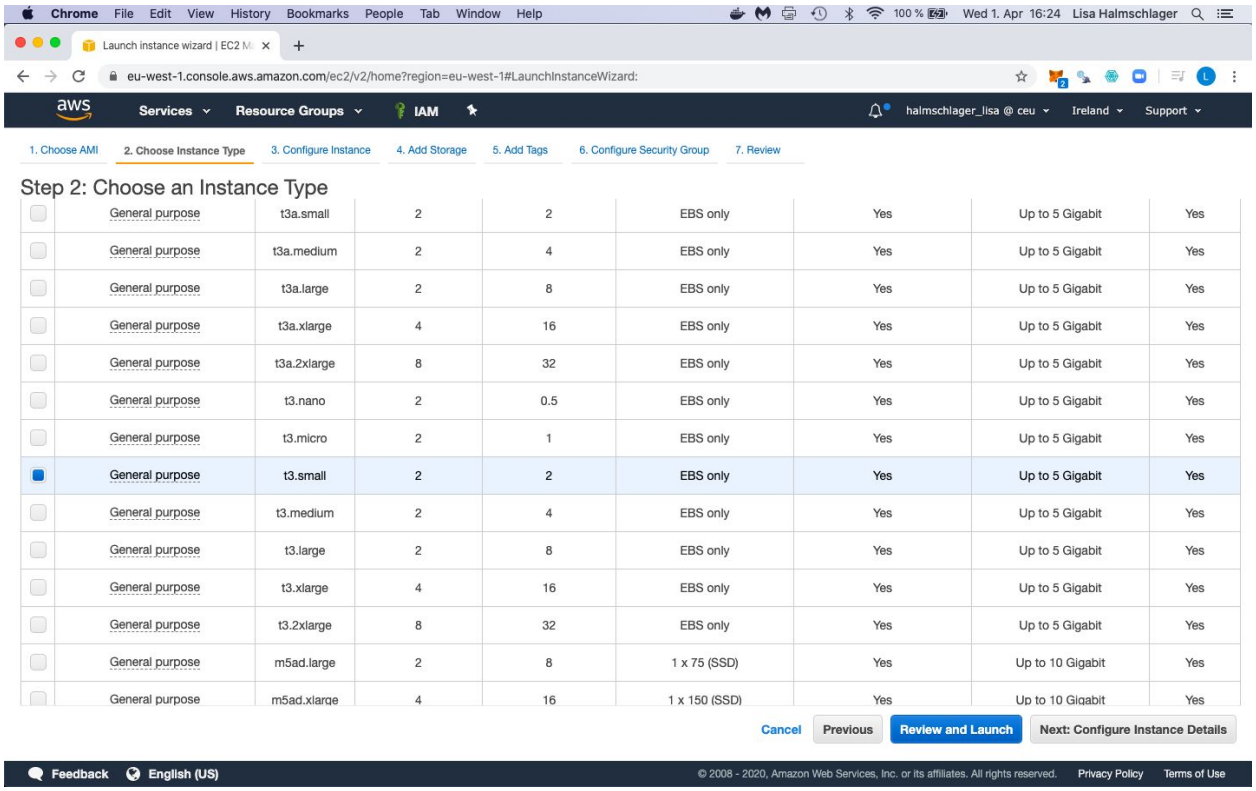
- I logged into the central **CEU AWS account**: https://ceu.signin.aws.amazon.com/console using 2FA.
- I used the **Ireland** region
- I went to the **EC2** console and created/launched a new t3.small instance using the de4-week3 **Amazon Machine Image (AMI)**, the gergely-week2 **EC2 IAM role**, and a new **security group** with the name DE4-1902224-sc where I opened up the 22 (ssh), 80 (web), 8000 (alternate ssh), 8787 (rstudio) and 8080 (jenkins) ports.

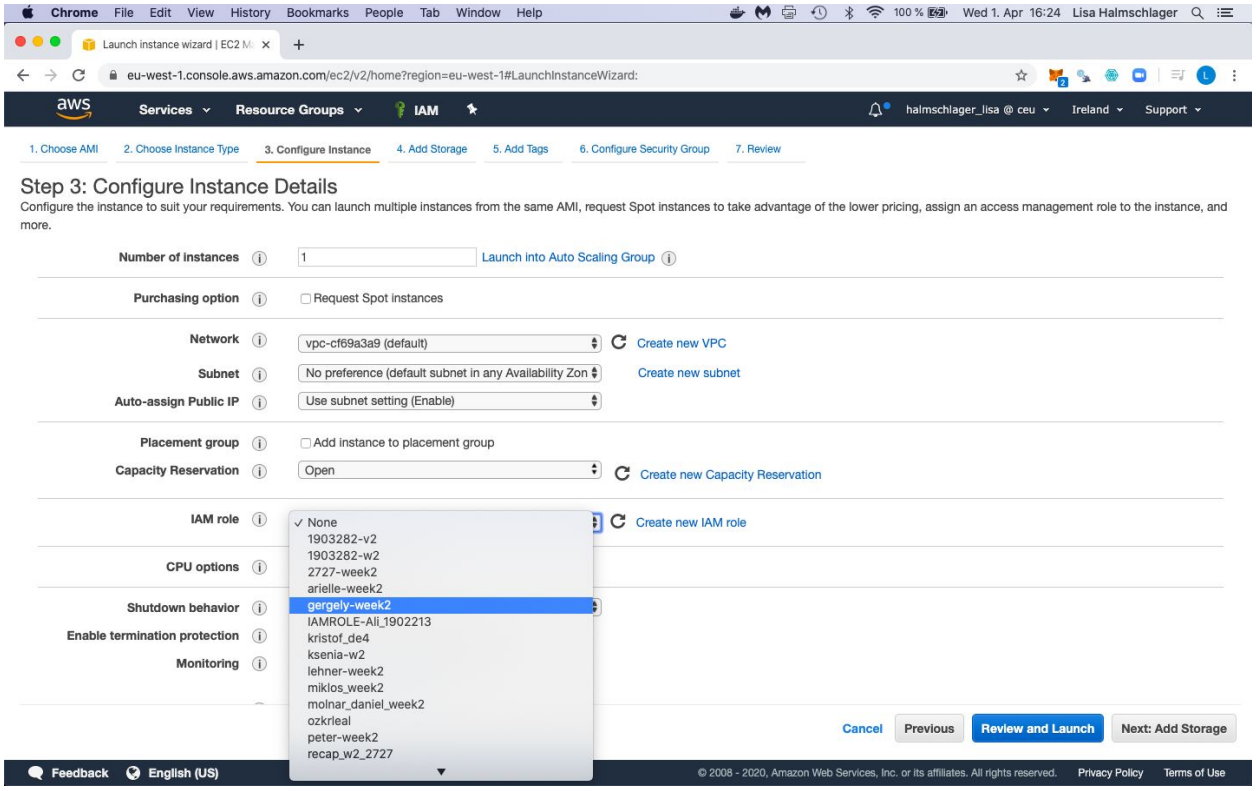
The de4-week3 Amazon Machine Image allows me to spin up an EC2 node with RStudio Server, Shiny Server, Jenkins, Redis and Docker installed & pre-configured along with the most often used R packages (including the ones we used for stream processing, eg botor, AWR.Kinesis and the binancer package).

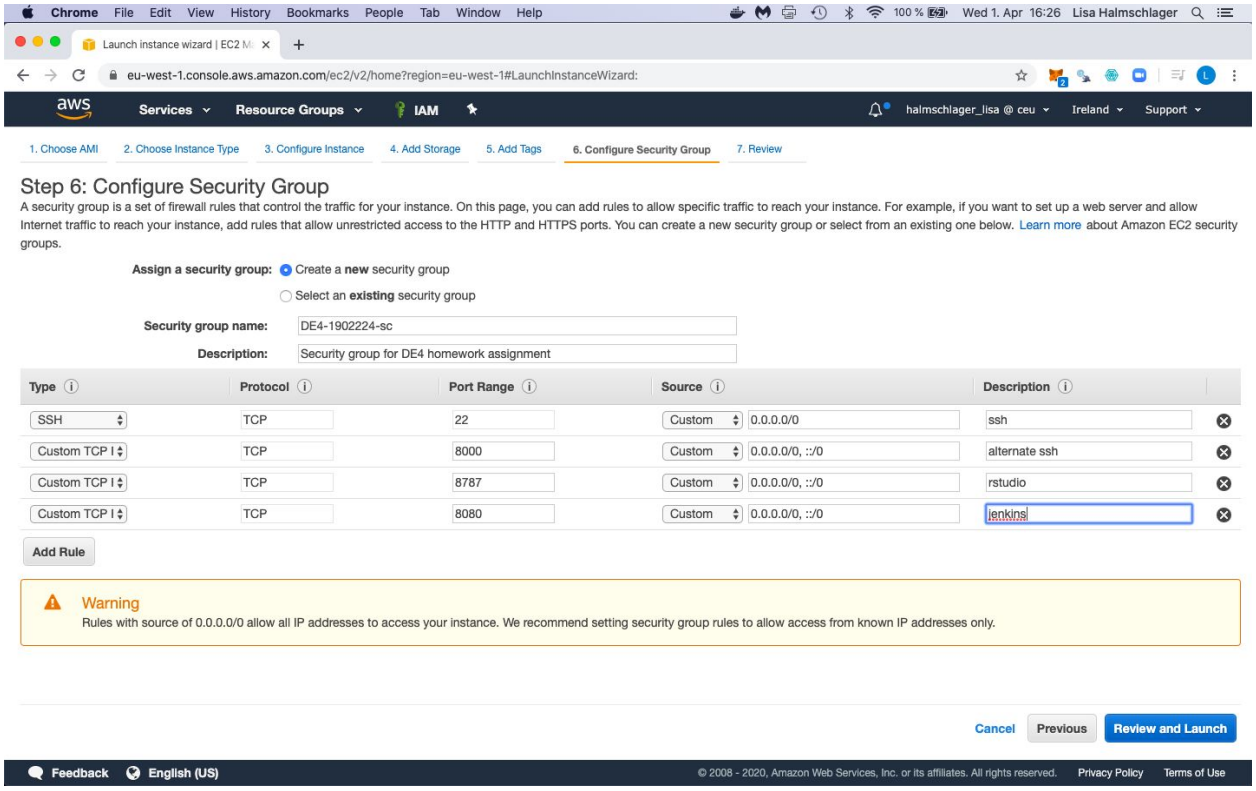
The gergely-week2 EC2 IAM role gives me full access to Kinesis, Dynamodb, Cloudwatch and encrypt/decrypt access to the "all-the-keys" **KMS** key.

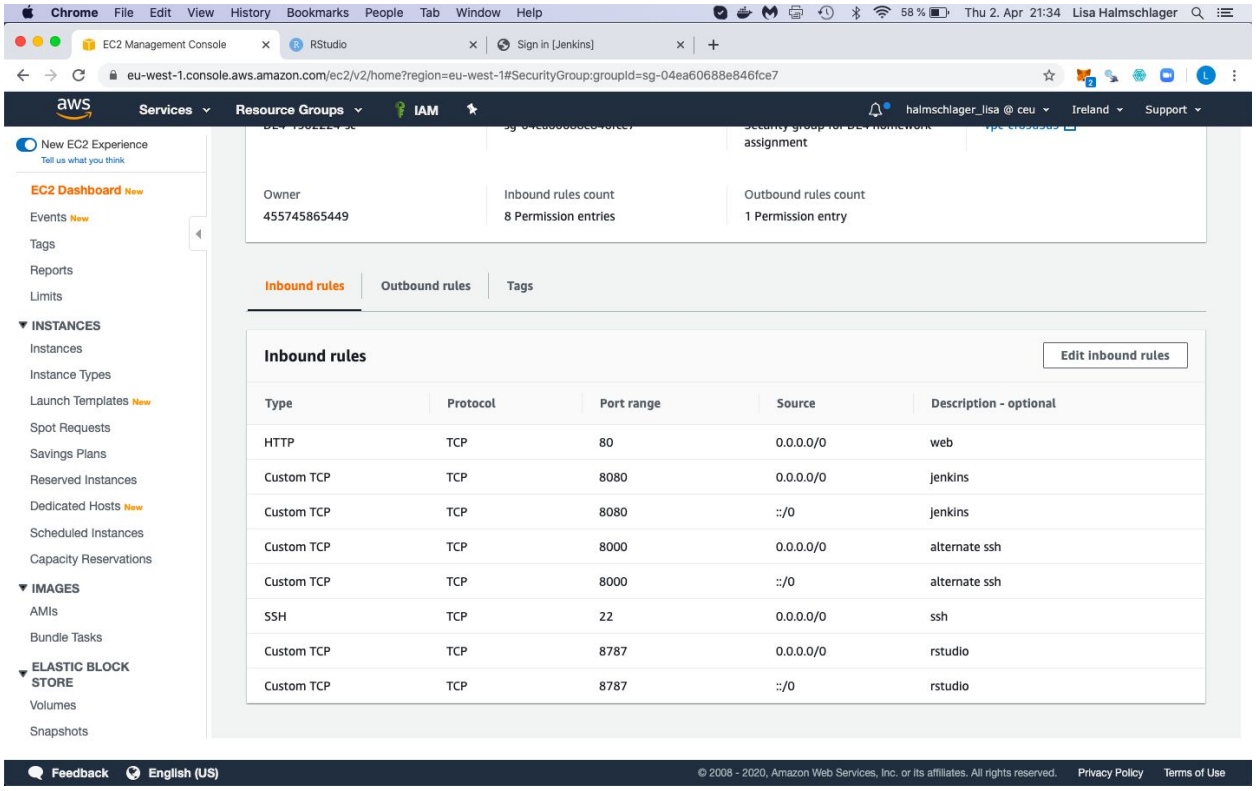


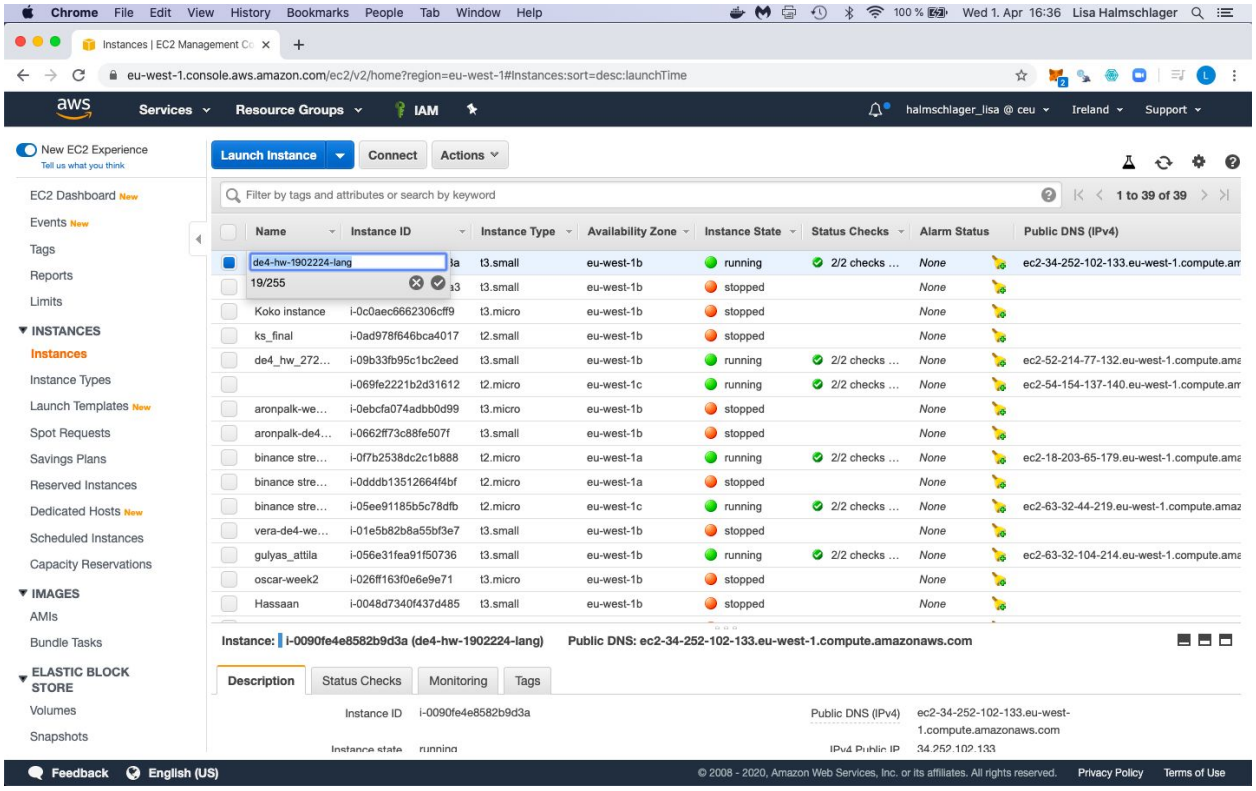

Preview File Edit View Go Tools Window Help
Wed 1. Apr 16:22 Lisa Halmschlager
Launch instance wizard | EC2
eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:
Services Resource Groups IAM
halmschlager_lisa @ ceu Ireland Support
1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review
Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Search for an AMI by entering a search term e.g. "Windows"
Quick Start
My AMIs
AWS Marketplace
Community AMIs
Ownership
Owned by me
Shared with me
Architecture
32-bit (x86)
64-bit (x86)
64-bit (Arm)
Root device type
1 to 4 of 4 AMIs
stock-ubuntu-18_04-but-ssh-on-port-8000 - ami-05e29314ed71317fa
running SSH on port 8000 as well
Root device type: ebs Virtualization type: hvm Owner: 455745865449 ENA Enabled: Yes
Select
64-bit (x86)
de4-week3 - ami-071c5ca9e5c81aed1
Root device type: ebs Virtualization type: hvm Owner: 455745865449 ENA Enabled: Yes
Select
64-bit (x86)
Ubuntu - SSH on port 8787 - ami-083d3d7f15e5cdcd3
Ubuntu image for CEU where the SSH server listens on port 8787
Root device type: ebs Virtualization type: hvm Owner: 455745865449 ENA Enabled: Yes
Select
64-bit (x86)
de4-week2 - ami-0ab0a49578237fe0c
Initial image for the 2nd week of Data Engineering 4 (with RStudio Server, Jenkins, Redis, Python and R packages installed)
Root device type: ebs Virtualization type: hvm Owner: 455745865449 ENA Enabled: Yes
Select
64-bit (x86)
Feedback English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy Terms of Use


Chrome File Edit View History Bookmarks People Tab Window Help
Wed 1. Apr 16:24 Lisa Halmschlager
Launch instance wizard | EC2
eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:
Services Resource Groups IAM
halmschlager_lisa @ ceu Ireland Support
1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review
Step 2: Choose an Instance Type
General purpose t3a.small 2 2 EBS only Yes Up to 5 Gigabit Yes
General purpose t3a.medium 2 4 EBS only Yes Up to 5 Gigabit Yes
General purpose t3a.large 2 8 EBS only Yes Up to 5 Gigabit Yes
General purpose t3a.xlarge 4 16 EBS only Yes Up to 5 Gigabit Yes
General purpose t3a.2xlarge 8 32 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.nano 2 0.5 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.micro 2 1 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.small 2 2 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.medium 2 4 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.large 2 8 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.xlarge 4 16 EBS only Yes Up to 5 Gigabit Yes
General purpose t3.2xlarge 8 32 EBS only Yes Up to 5 Gigabit Yes
General purpose m5ad.large 2 8 1 x 75 (SSD) Yes Up to 10 Gigabit Yes
General purpose m5ad.xlarge 4 16 1 x 150 (SSD) Yes Up to 10 Gigabit Yes
Cancel Previous Review and Launch Next: Configure Instance Details
Feedback English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy Terms of Use


Chrome File Edit View History Bookmarks People Tab Window Help
Wed 1. Apr 16:24 Lisa Halmschlager
Launch instance wizard | EC2
eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:
aws Services Resource Groups IAM halmschlager_lisa @ ceu Ireland Support
1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review
Step 3: Configure Instance Details
Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.
Number of instances 1 Launch into Auto Scaling Group
Purchasing option Request Spot instances
Network vpc-cf69a3a9 (default) Create new VPC
Subnet No preference (default subnet in any Availability Zon Create new subnet
Auto-assign Public IP Use subnet setting (Enable)
Placement group Add instance to placement group
Capacity Reservation Open Create new Capacity Reservation
IAM role Create new IAM role
CPU options
Shutdown behavior
Enable termination protection
Monitoring
None
1903282-v2
1903282-w2
2727-week2
arielle-week2
gergely-week2
IAMROLE-Ali_1902213
kristof_de4
ksenia-w2
lehner-week2
miklos_week2
molnar_daniel_week2
ozkrleal
peter-week2
recap_w2_2727
Cancel Previous Review and Launch Next: Add Storage
Feedback English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use


Chrome File Edit View History Bookmarks People Tab Window Help
Wed 1. Apr 16:26 Lisa Halmschlager
Launch instance wizard | EC2
eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:
aws Services Resource Groups IAM halmschlager_lisa @ ceu Ireland Support
1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review
Step 6: Configure Security Group
A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.
Assign a security group: Create a new security group
Select an existing security group
Security group name: DE4-1902224-sc
Description: Security group for DE4 homework assignment
Type Protocol Port Range Source Description
SSH TCP 22 Custom 0.0.0.0/0 ssh
Custom TCP I TCP 8000 Custom 0.0.0.0/0, ::/0 alternate ssh
Custom TCP I TCP 8787 Custom 0.0.0.0/0, ::/0 rstudio
Custom TCP I TCP 8080 Custom 0.0.0.0/0, ::/0 jenkins
Add Rule
Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Cancel Previous Review and Launch
Feedback English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

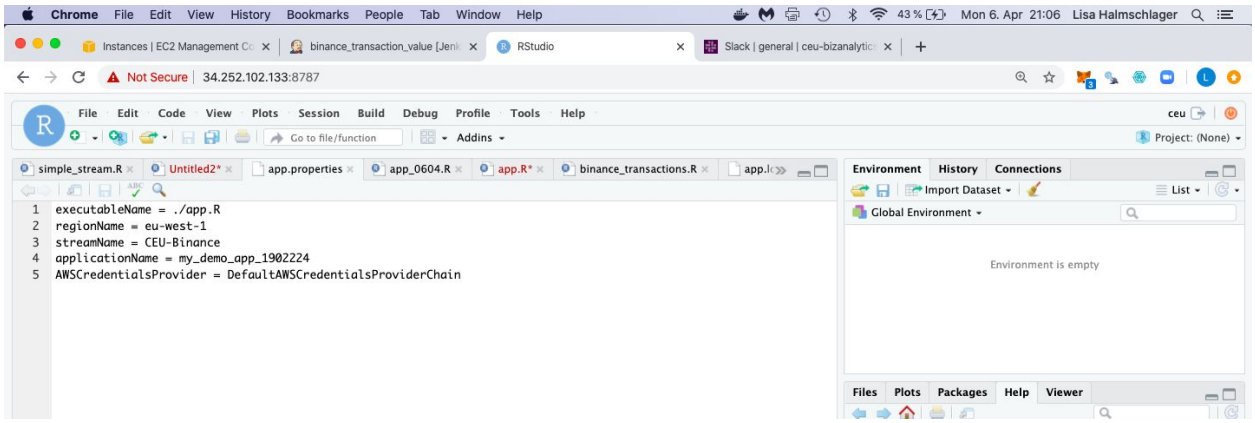

Chrome File Edit View History Bookmarks People Tab Window Help
Thu 2. Apr 21:34 Lisa Halmschlager
EC2 Management Console
RStudio
Sign in [Jenkins]
eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#SecurityGroup:groupId=sg-04ea60688e846fce7
Services
Resource Groups
IAM
halmschlager_lisa @ ceu
Ireland
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events New
Tags
Reports
Limits
INSTANCES
Instances
Instance Types
Launch Templates New
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
assignment
Owner
455745865449
Inbound rules count
8 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules
Edit inbound rules
Type
Protocol
Port range
Source
Description - optional
HTTP TCP 80 0.0.0.0/0 web
Custom TCP TCP 8080 0.0.0.0/0 jenkins
Custom TCP TCP 8080 ::/0 jenkins
Custom TCP TCP 8000 0.0.0.0/0 alternate ssh
Custom TCP TCP 8000 ::/0 alternate ssh
SSH TCP 22 0.0.0.0/0 ssh
Custom TCP TCP 8787 0.0.0.0/0 rstudio
Custom TCP TCP 8787 ::/0 rstudio
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

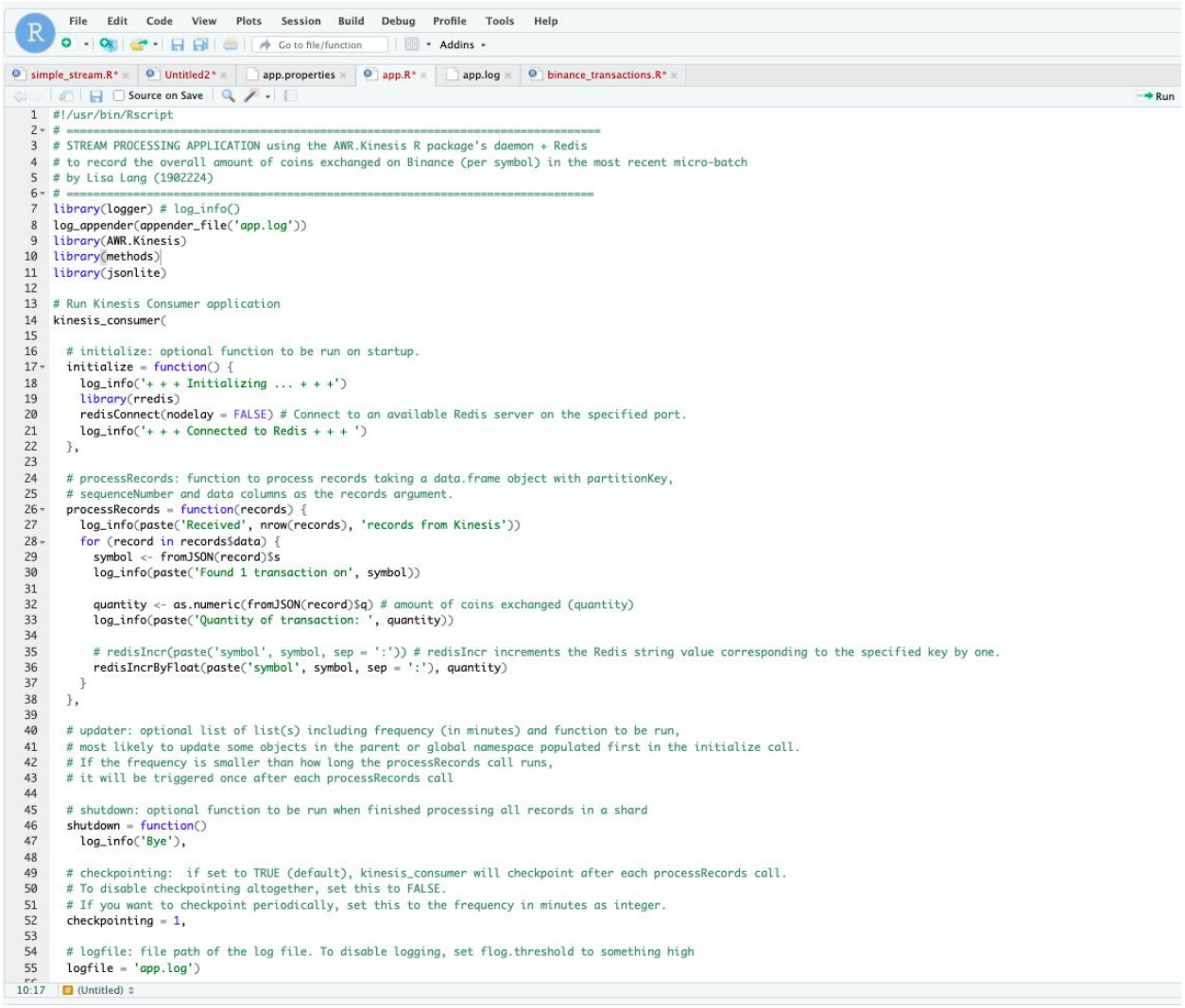

Chrome File Edit View History Bookmarks People Tab Window Help
Wed 1. Apr 16:36 Lisa Halmschlager
Instances | EC2 Management C
eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#Instances:sort=desc:launchTime
Services
Resource Groups
IAM
halmschlager_lisa @ ceu
Ireland
Support
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 39 of 39
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
de4-hw-1902224-lang t3.small eu-west-1b running 2/2 checks ... None ec2-34-252-102-133.eu-west-1.compute.am
19/255 t3.small eu-west-1b stopped None
Koko instance i-0c0aec6662306cff9 t3.micro eu-west-1b stopped None
ks_final i-0ad978f646bca4017 t2.small eu-west-1b stopped None
de4_hw_272... i-09b33fb95c1bc2eed t3.small eu-west-1b running 2/2 checks ... None ec2-52-214-77-132.eu-west-1.compute.ama
i-069fe2221b2d31612 t2.micro eu-west-1c running 2/2 checks ... None ec2-54-154-137-140.eu-west-1.compute.am
aronpalk-we... i-0ebcfa074adbb0d99 t3.micro eu-west-1b stopped None
aronpalk-de4... i-0662ff73c88fe507f t3.small eu-west-1b stopped None
binance stre... i-0f7b2538dc2c1b888 t2.micro eu-west-1a running 2/2 checks ... None ec2-18-203-65-179.eu-west-1.compute.ama
binance stre... i-0dddb13512664f4bf t2.micro eu-west-1a stopped None
binance stre... i-05ee91185b5c78dfb t2.micro eu-west-1c running 2/2 checks ... None ec2-63-32-44-219.eu-west-1.compute.amaz
vera-de4-we... i-01e5b82b8a55bf3e7 t3.small eu-west-1b stopped None
gulyas_attila i-056e31fea91f50736 t3.small eu-west-1b running 2/2 checks ... None ec2-63-32-104-214.eu-west-1.compute.ama
oscar-week2 i-026ff163f0e6e9e71 t3.micro eu-west-1b stopped None
Hassaan i-0048d7340f437d485 t3.small eu-west-1b stopped None
Instance: i-0090fe4e8582b9d3a (de4-hw-1902224-lang) Public DNS: ec2-34-252-102-133.eu-west-1.compute.amazonaws.com
Description
Status Checks
Monitoring
Tags
Instance ID i-0090fe4e8582b9d3a
Public DNS (IPv4) ec2-34-252-102-133.eu-west-1.compute.amazonaws.com
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

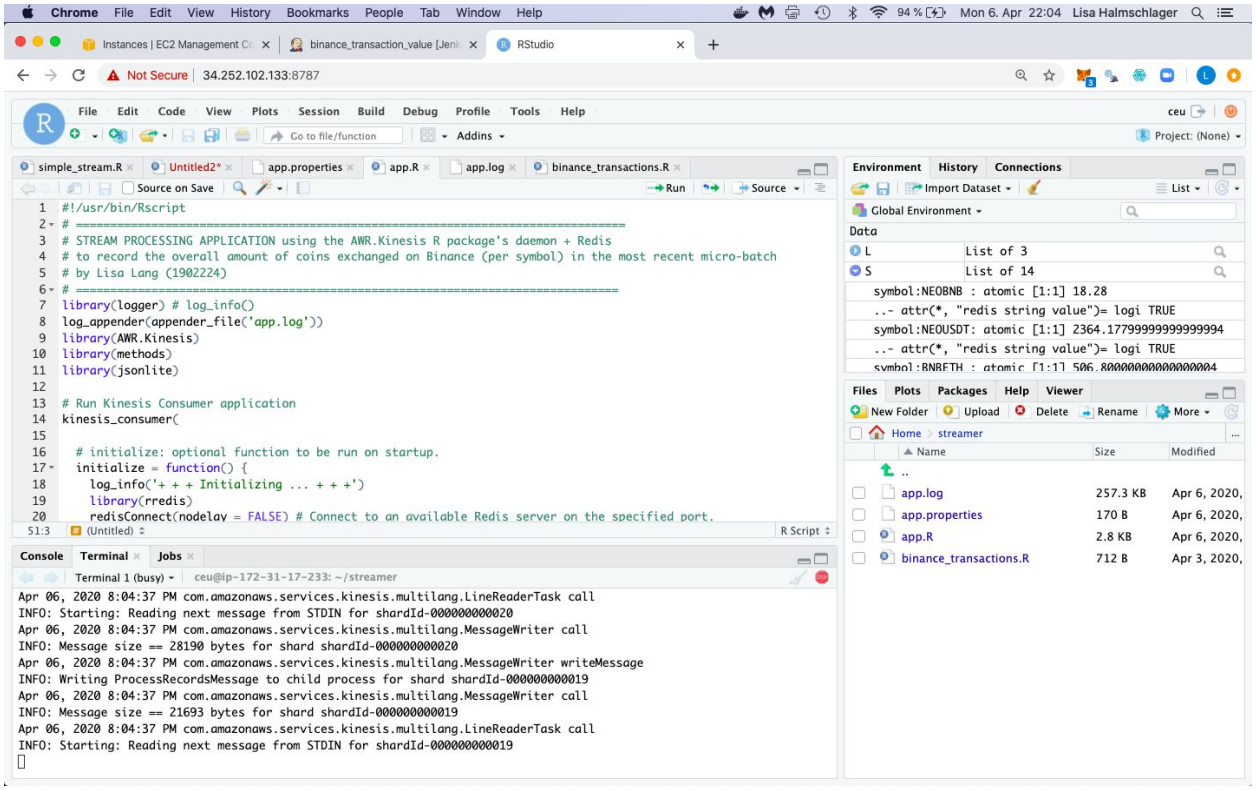
R code in my RStudio Server

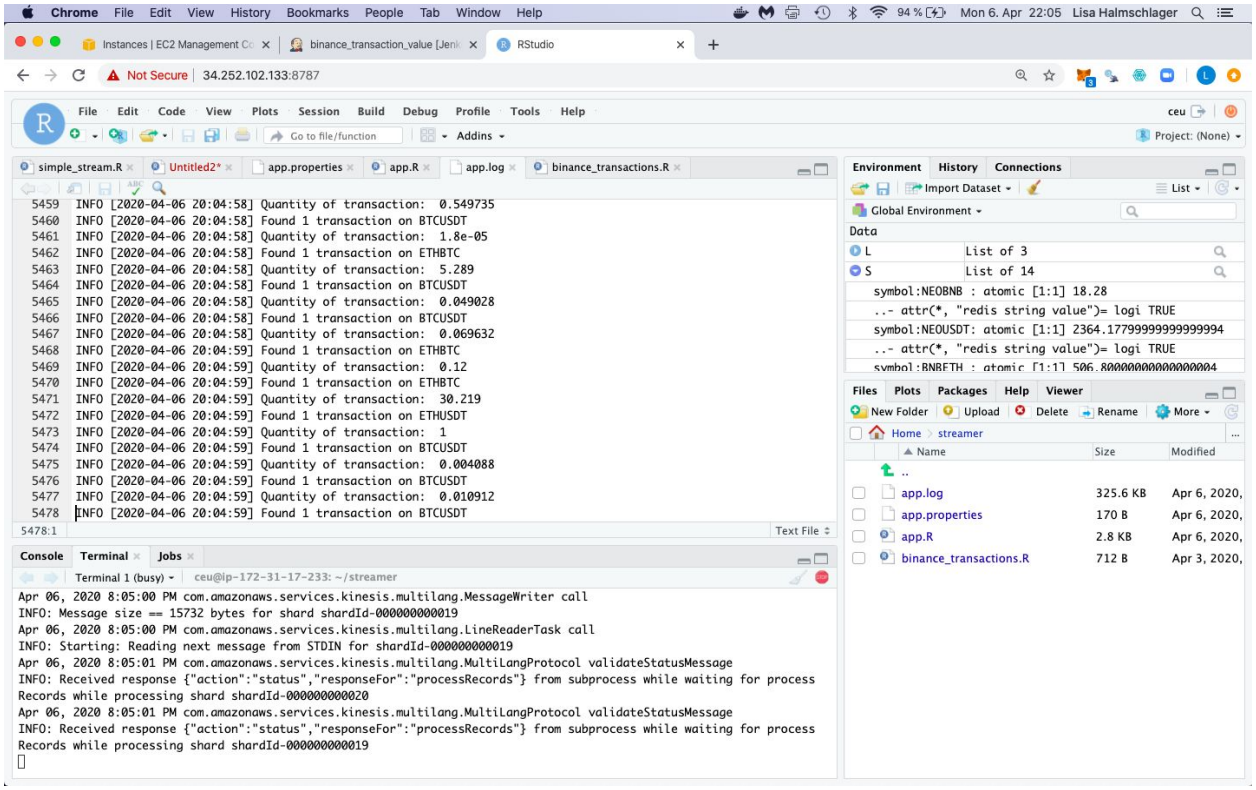
I created three R scripts to get some data from the stream via the **AWS Java SDK** that interacts with our Kinesis stream “CEU-Binance” and to process it further.

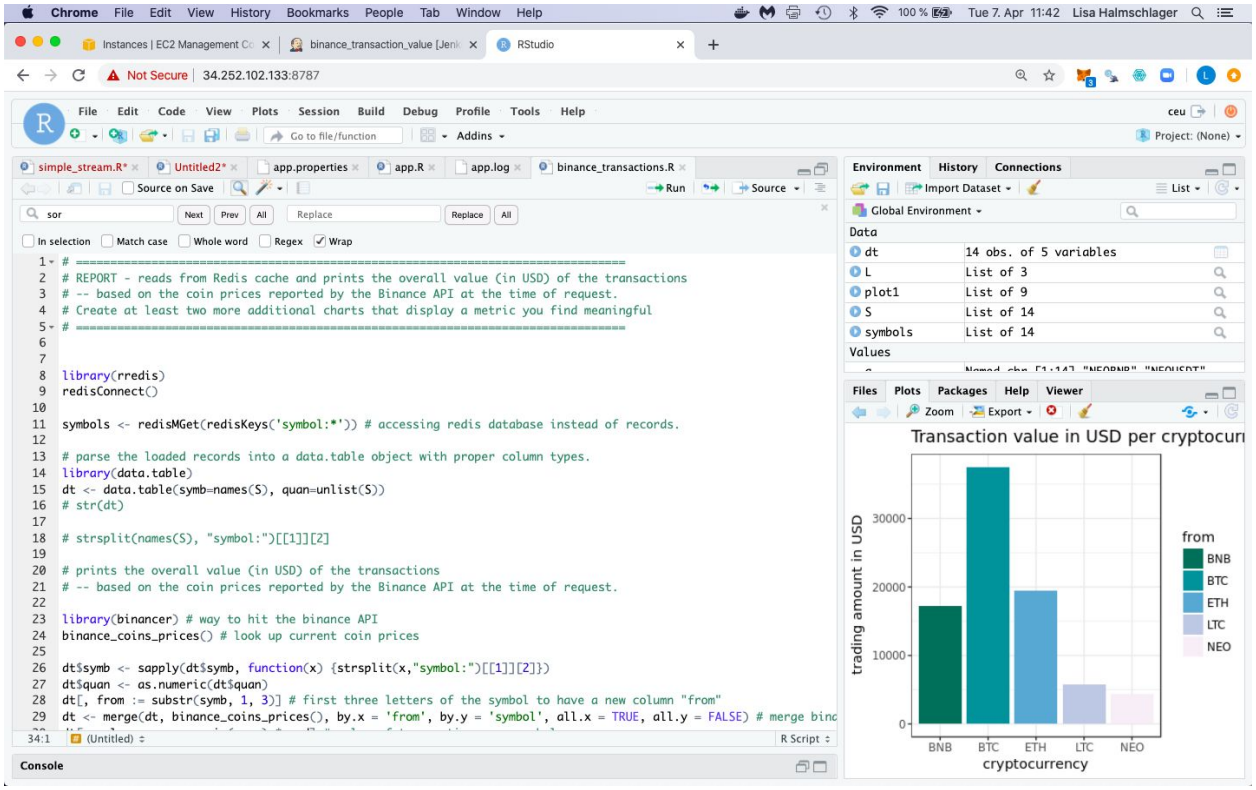
- I logged in to **RStudio** using the new instance's public IP address (34.252.102.133) and 8787 port and the given credentials.
- I created a new folder for the scripts: streamer.
- I created a new text file within that subfolder called app.properties, that includes code to store the configurable parameters of my application.
- I created an R script within that subfolder called app.R that reads data from the CEU-Binance stream and stores it in a **Redis Database** (key-value pair). The quantities recorded are incremented with every new read.
- I converted the R script into an executable using the Terminal.
- Then I ran the app in the Terminal.
- I create an R script called binance_transactions.R that reads in the quantities of transactions from the Redis cache and prints the overall value (in USD) of the transactions, based on the coin prices reported by the Binance API at the time of request. It also creates two charts : A bar chart showing the transaction value in USD per cryptocurrency and a bar chart showing the transaction value per currency-pair.
  The value of transactions and bar charts are sent to #bots-final-project **slack channel**.

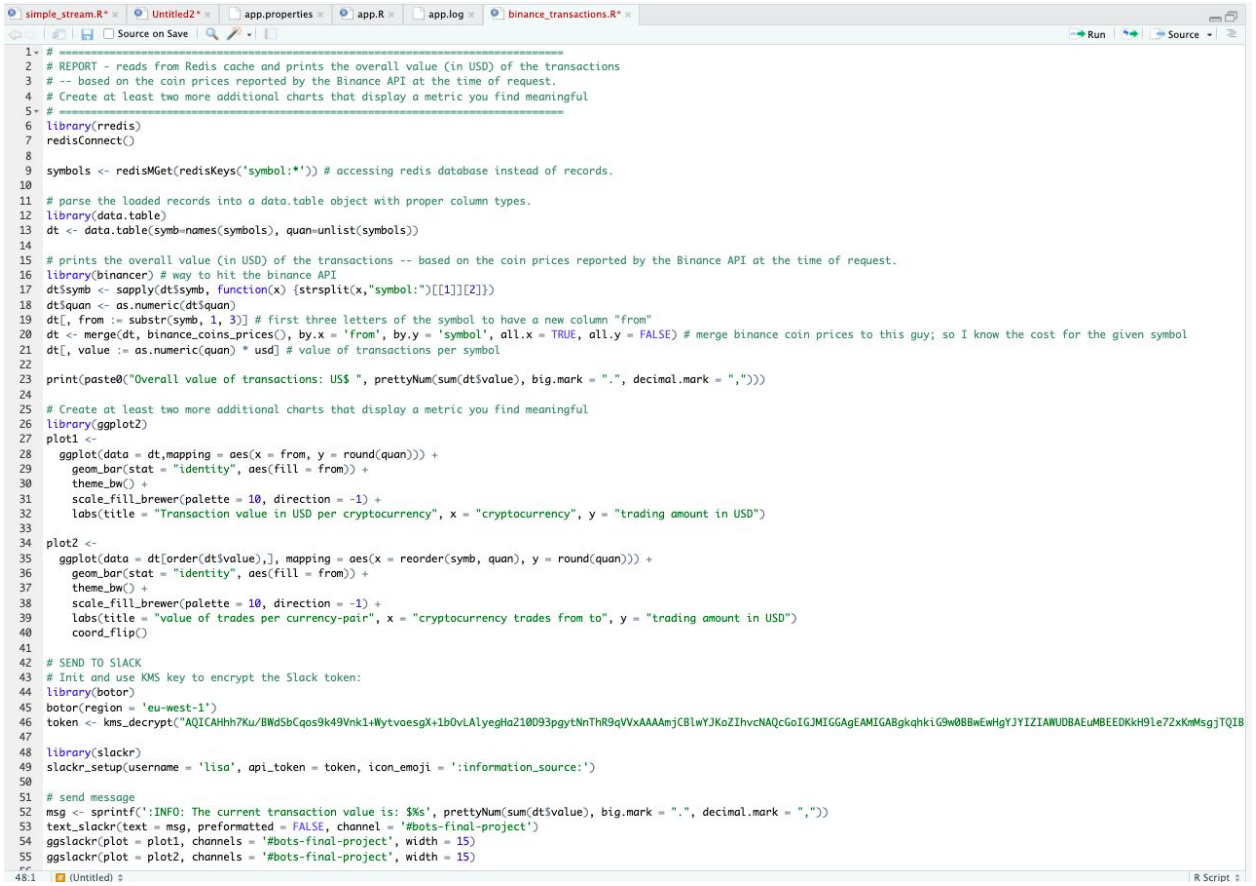



File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
simple_stream.R*
Untitled2*
app.properties
app.R*
app.log
binance_transactions.R*
Source on Save
Run
#!/usr/bin/Rscript
# ==
# STREAM PROCESSING APPLICATION using the AWR.Kinesis R package's daemon + Redis
# to record the overall amount of coins exchanged on Binance (per symbol) in the most recent micro-batch
# by Lisa Lang (1902224)
# ==
library(logger) # log_info()
log_appender(appender_file('app.log'))
library(AWR.Kinesis)
library(methods)
library(jsonlite)
# Run Kinesis Consumer application
kinesis_consumer(
# initialize: optional function to be run on startup.
initialize = function() {
log_info('+ + + Initializing ... + + +')
library(rredis)
redisConnect(nodelay = FALSE) # Connect to an available Redis server on the specified port.
log_info('+ + + Connected to Redis + + + ')
},
# processRecords: function to process records taking a data.frame object with partitionKey,
# sequenceNumber and data columns as the records argument.
processRecords = function(records) {
log_info(paste('Received', nrow(records), 'records from Kinesis'))
for (record in records$data) {
symbol <- fromJSON(record)$s
log_info(paste('Found 1 transaction on', symbol))
quantity <- as.numeric(fromJSON(record)$q) # amount of coins exchanged (quantity)
log_info(paste('Quantity of transaction: ', quantity))
# redisIncr(paste('symbol', symbol, sep = ':')) # redisIncr increments the Redis string value corresponding to the specified key by one.
redisIncrByFloat(paste('symbol', symbol, sep = ':'), quantity)
}
},
# updater: optional list of list(s) including frequency (in minutes) and function to be run,
# most likely to update some objects in the parent or global namespace populated first in the initialize call.
# If the frequency is smaller than how long the processRecords call runs,
# it will be triggered once after each processRecords call
# shutdown: optional function to be run when finished processing all records in a shard
shutdown = function()
log_info('Bye'),
# checkpointing: if set to TRUE (default), kinesis_consumer will checkpoint after each processRecords call.
# To disable checkpointing altogether, set this to FALSE.
# If you want to checkpoint periodically, set this to the frequency in minutes as integer.
checkpointing = 1,
# logfile: file path of the log file. To disable logging, set flog.threshold to something high
logfile = 'app.log')
10:17
(Untitled)


Not Secure 34.252.102.133:8787
simple_stream.R
Untitled2*
app.properties
app.R
app.log
binance_transactions.R
#!/usr/bin/Rscript
# ==========
# STREAM PROCESSING APPLICATION using the AWR.Kinesis R package's daemon + Redis
# to record the overall amount of coins exchanged on Binance (per symbol) in the most recent micro-batch
# by Lisa Lang (1902224)
# ==========
library(logger) # log_info()
log_appender(appender_file('app.log'))
library(AWR.Kinesis)
library(methods)
library(jsonlite)
# Run Kinesis Consumer application
kinesis_consumer(
# initialize: optional function to be run on startup.
initialize = function() {
log_info('+ + + Initializing ... + + +')
library(rredis)
redisConnect(nodelay = FALSE) # Connect to an available Redis server on the specified port.
Environment
History
Connections
Global Environment
Data
L List of 3
S List of 14
symbol:NEOBNB : atomic [1:1] 18.28
..- attr(*, "redis string value")= logi TRUE
symbol:NEOUSDT: atomic [1:1] 2364.17799999999999994
..- attr(*, "redis string value")= logi TRUE
Files
Plots
Packages
Help
Viewer
New Folder
Upload
Delete
Rename
More
Home
streamer
Name
Size
Modified
app.log 257.3 KB Apr 6, 2020,
app.properties 170 B Apr 6, 2020,
app.R 2.8 KB Apr 6, 2020,
binance_transactions.R 712 B Apr 3, 2020,
Console
Terminal
Jobs
Apr 06, 2020 8:04:37 PM com.amazonaws.services.kinesis.multilang.LineReaderTask call
INFO: Starting: Reading next message from STDIN for shardId-000000000020
Apr 06, 2020 8:04:37 PM com.amazonaws.services.kinesis.multilang.MessageWriter call
INFO: Message size == 28190 bytes for shard shardId-000000000020
Apr 06, 2020 8:04:37 PM com.amazonaws.services.kinesis.multilang.MessageWriter writeMessage
INFO: Writing ProcessRecordsMessage to child process for shard shardId-000000000019
Apr 06, 2020 8:04:37 PM com.amazonaws.services.kinesis.multilang.MessageWriter call
INFO: Message size == 21693 bytes for shard shardId-000000000019
Apr 06, 2020 8:04:37 PM com.amazonaws.services.kinesis.multilang.LineReaderTask call
INFO: Starting: Reading next message from STDIN for shardId-000000000019

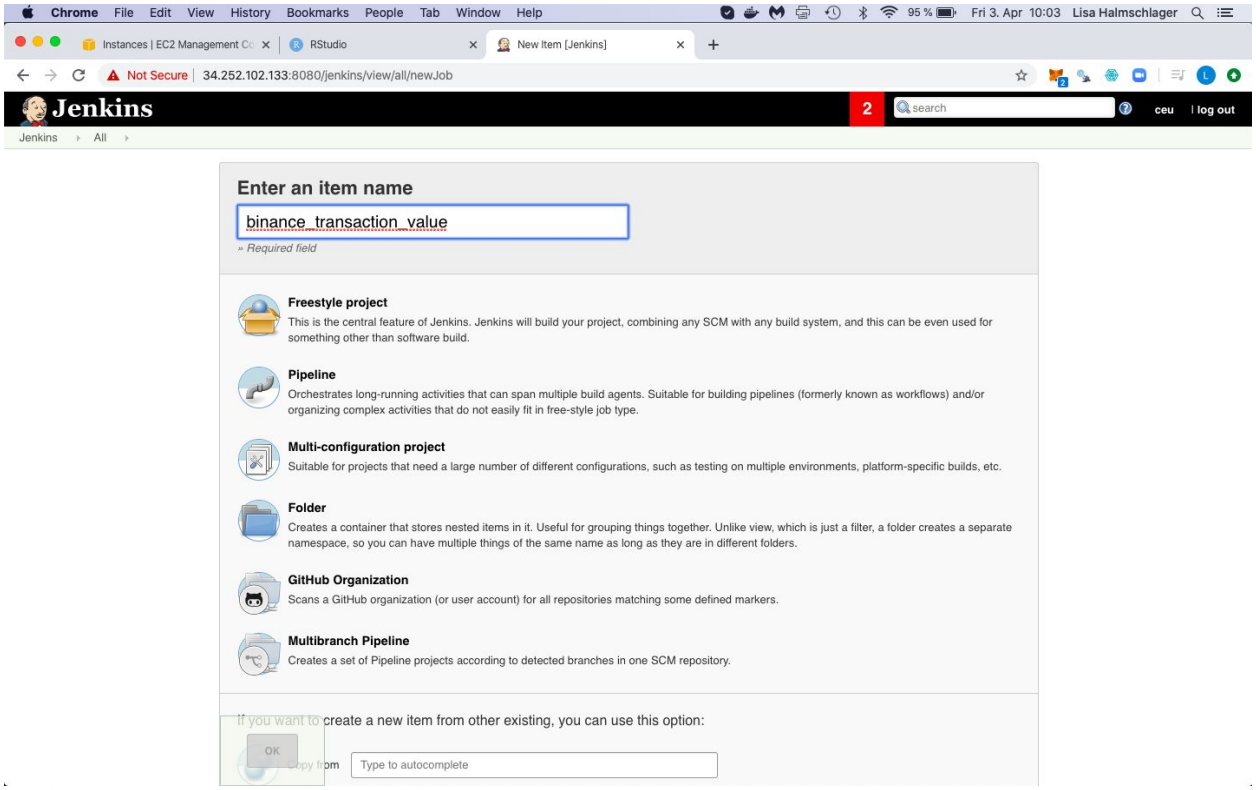

Not Secure 34.252.102.133:8787
simple_stream.R
Untitled2*
app.properties
app.R
app.log
binance_transactions.R
5459 INFO [2020-04-06 20:04:58] Quantity of transaction: 0.549735
5460 INFO [2020-04-06 20:04:58] Found 1 transaction on BTCUSDT
5461 INFO [2020-04-06 20:04:58] Quantity of transaction: 1.8e-05
5462 INFO [2020-04-06 20:04:58] Found 1 transaction on ETHBTC
5463 INFO [2020-04-06 20:04:58] Quantity of transaction: 5.289
5464 INFO [2020-04-06 20:04:58] Found 1 transaction on BTCUSDT
5465 INFO [2020-04-06 20:04:58] Quantity of transaction: 0.049028
5466 INFO [2020-04-06 20:04:58] Found 1 transaction on BTCUSDT
5467 INFO [2020-04-06 20:04:58] Quantity of transaction: 0.069632
5468 INFO [2020-04-06 20:04:59] Found 1 transaction on ETHBTC
5469 INFO [2020-04-06 20:04:59] Quantity of transaction: 0.12
5470 INFO [2020-04-06 20:04:59] Found 1 transaction on ETHBTC
5471 INFO [2020-04-06 20:04:59] Quantity of transaction: 30.219
5472 INFO [2020-04-06 20:04:59] Found 1 transaction on ETHUSDT
5473 INFO [2020-04-06 20:04:59] Quantity of transaction: 1
5474 INFO [2020-04-06 20:04:59] Found 1 transaction on BTCUSDT
5475 INFO [2020-04-06 20:04:59] Quantity of transaction: 0.004088
5476 INFO [2020-04-06 20:04:59] Found 1 transaction on BTCUSDT
5477 INFO [2020-04-06 20:04:59] Quantity of transaction: 0.010912
5478 INFO [2020-04-06 20:04:59] Found 1 transaction on BTCUSDT
5478:1
Text File
Environment
History
Connections
Global Environment
Data
L List of 3
S List of 14
symbol:NEOBNB : atomic [1:1] 18.28
..- attr(*, "redis string value")= logi TRUE
symbol:NEOUSDT: atomic [1:1] 2364.17799999999999994
..- attr(*, "redis string value")= logi TRUE
Files
Plots
Packages
Help
Viewer
New Folder
Upload
Delete
Rename
More
Home
streamer
Name
Size
Modified
app.log 325.6 KB Apr 6, 2020,
app.properties 170 B Apr 6, 2020,
app.R 2.8 KB Apr 6, 2020,
binance_transactions.R 712 B Apr 3, 2020,
Console
Terminal
Jobs
Apr 06, 2020 8:05:00 PM com.amazonaws.services.kinesis.multilang.MessageWriter call
INFO: Message size == 15732 bytes for shard shardId-000000000019
Apr 06, 2020 8:05:00 PM com.amazonaws.services.kinesis.multilang.LineReaderTask call
INFO: Starting: Reading next message from STDIN for shardId-000000000019
Apr 06, 2020 8:05:01 PM com.amazonaws.services.kinesis.multilang.MultiLangProtocol validateStatusMessage
INFO: Received response {"action":"status","responseFor":"processRecords"} from subprocess while waiting for process
Records while processing shard shardId-000000000020
Apr 06, 2020 8:05:01 PM com.amazonaws.services.kinesis.multilang.MultiLangProtocol validateStatusMessage
INFO: Received response {"action":"status","responseFor":"processRecords"} from subprocess while waiting for process
Records while processing shard shardId-000000000019

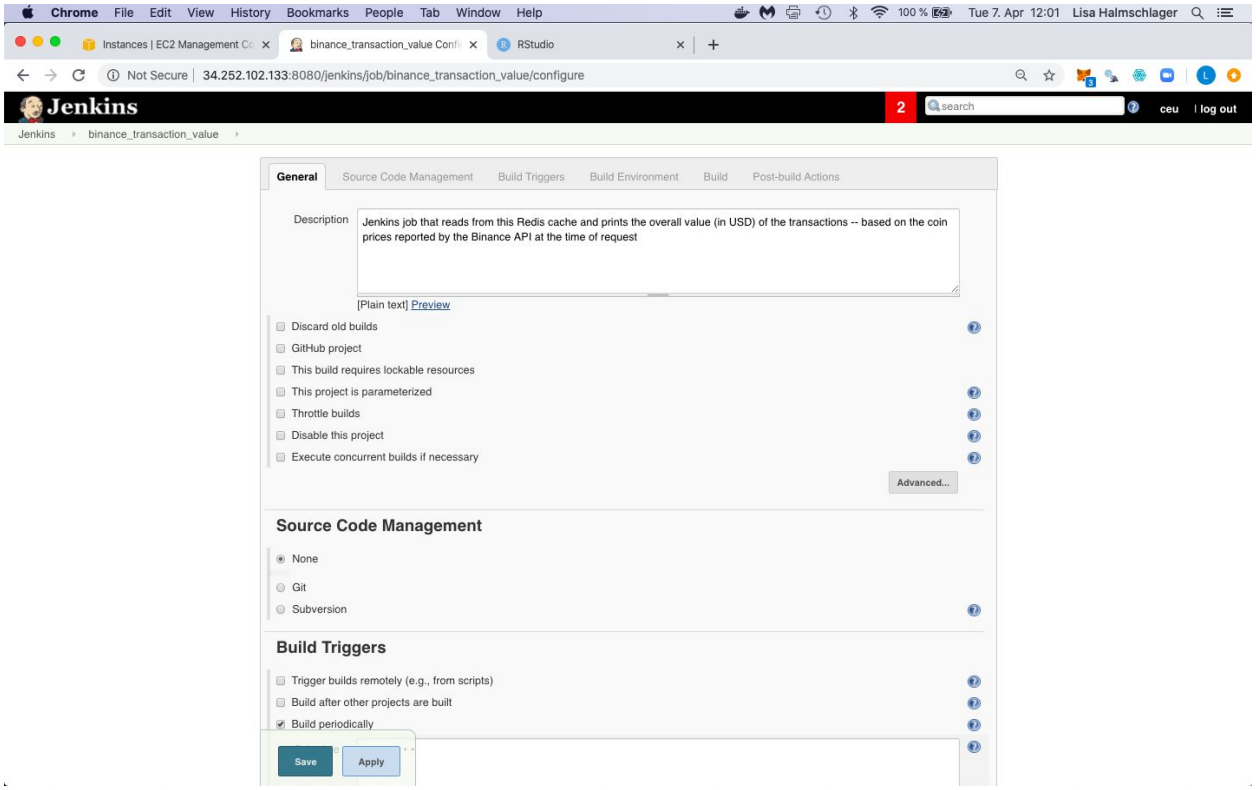

Chrome File Edit View History Bookmarks People Tab Window Help
Tue 7. Apr 11:42 Lisa Halmschlager
Not Secure | 34.252.102.133:8787
File Edit Code View Plots Session Build Debug Profile Tools Help
simple_stream.R* Untitled2* app.properties app.R app.log binance_transactions.R
Source on Save
Run Source
In selection Match case Whole word Regex Wrap
1 # ==========
2 # REPORT - reads from Redis cache and prints the overall value (in USD) of the transactions
3 # -- based on the coin prices reported by the Binance API at the time of request.
4 # Create at least two more additional charts that display a metric you find meaningful
5 # ==========
6
7
8 library(rredis)
9 redisConnect()
10
11 symbols <- redisMGet(redisKeys('symbol:*')) # accessing redis database instead of records.
12
13 # parse the loaded records into a data.table object with proper column types.
14 library(data.table)
15 dt <- data.table(symb=names(S), quan=unlist(S))
16 # str(dt)
17
18 # strsplit(names(S), "symbol:")[[1]][2]
19
20 # prints the overall value (in USD) of the transactions
21 # -- based on the coin prices reported by the Binance API at the time of request.
22
23 library(binancer) # way to hit the binance API
24 binance_coins_prices() # look up current coin prices
25
26 dt$symb <- sapply(dt$symb, function(x) {strsplit(x,"symbol:")[[1]][2]})
27 dt$quan <- as.numeric(dt$quan)
28 dt[, from := substr(symb, 1, 3)] # first three letters of the symbol to have a new column "from"
29 dt <- merge(dt, binance_coins_prices(), by.x = 'from', by.y = 'symbol', all.x = TRUE, all.y = FALSE) # merge binc
34:1 (Untitled) R Script
Console
Environment History Connections
Import Dataset List
Global Environment
Data
dt 14 obs. of 5 variables
L List of 3
plot1 List of 9
S List of 14
symbols List of 14
Values
Files Plots Packages Help Viewer
Zoom Export
Transaction value in USD per cryptocurr
trading amount in USD
30000
20000
10000
0
BNB BTC ETH LTC NEO
cryptocurrency
from
BNB
BTC
ETH
LTC
NEO


simple_stream.R* Untitled2* app.properties app.R app.log binance_transactions.R*
Source on Save
Run Source
1 # ==========
2 # REPORT - reads from Redis cache and prints the overall value (in USD) of the transactions
3 # -- based on the coin prices reported by the Binance API at the time of request.
4 # Create at least two more additional charts that display a metric you find meaningful
5 # ==========
6 library(rredis)
7 redisConnect()
8
9 symbols <- redisMGet(redisKeys('symbol:*')) # accessing redis database instead of records.
10
11 # parse the loaded records into a data.table object with proper column types.
12 library(data.table)
13 dt <- data.table(symb=names(symbols), quan=unlist(symbols))
14
15 # prints the overall value (in USD) of the transactions -- based on the coin prices reported by the Binance API at the time of request.
16 library(binancer) # way to hit the binance API
17 dt$symb <- sapply(dt$symb, function(x) {strsplit(x,"symbol:")[[1]][2]})
18 dt$quan <- as.numeric(dt$quan)
19 dt[, from := substr(symb, 1, 3)] # first three letters of the symbol to have a new column "from"
20 dt <- merge(dt, binance_coins_prices(), by.x = 'from', by.y = 'symbol', all.x = TRUE, all.y = FALSE) # merge binance coin prices to this guy; so I know the cost for the given symbol
21 dt[, value := as.numeric(quan) * usd] # value of transactions per symbol
22
23 print(paste0("Overall value of transactions: US$ ", prettyNum(sum(dt$value), big.mark = ".", decimal.mark = ",")))
24
25 # Create at least two more additional charts that display a metric you find meaningful
26 library(ggplot2)
27 plot1 <-
28 ggplot(data = dt,mapping = aes(x = from, y = round(quan))) +
29 geom_bar(stat = "identity", aes(fill = from)) +
30 theme_bw() +
31 scale_fill_brewer(palette = 10, direction = -1) +
32 labs(title = "Transaction value in USD per cryptocurrency", x = "cryptocurrency", y = "trading amount in USD")
33
34 plot2 <-
35 ggplot(data = dt[order(dt$value),], mapping = aes(x = reorder(symb, quan), y = round(quan))) +
36 geom_bar(stat = "identity", aes(fill = from)) +
37 theme_bw() +
38 scale_fill_brewer(palette = 10, direction = -1) +
39 labs(title = "value of trades per currency-pair", x = "cryptocurrency trades from to", y = "trading amount in USD")
40 coord_flip()
41
42 # SEND TO SLACK
43 # Init and use KMS key to encrypt the Slack token:
44 library(botor)
45 botor(region = 'eu-west-1')
46 token <- kms_decrypt("AQICAHhh7Ku/BWdSbCqos9k49Vnk1+WytvoesgX+1bOvLAlyegHa210D93pgytNnThR9qVVxAAAAmjCBlwYJKoZIhvcNAQcGoIGJMIGGAgEAMIGABgkqhkiG9w0BBwEwHgYJYIZIAWUDBAEuMBEEDKkH9le72xKmMsgjTQIB
47
48 library(slackr)
49 slackr_setup(username = 'lisa', api_token = token, icon_emoji = ':information_source:')
50
51 # send message
52 msg <- sprintf(':INFO: The current transaction value is: $%s', prettyNum(sum(dt$value), big.mark = ".", decimal.mark = ","))
53 text_slackr(text = msg, preformatted = FALSE, channel = '#bots-final-project')
54 ggslackr(plot = plot1, channels = '#bots-final-project', width = 15)
55 ggslackr(plot = plot2, channels = '#bots-final-project', width = 15)
48:1 (Untitled) R Script

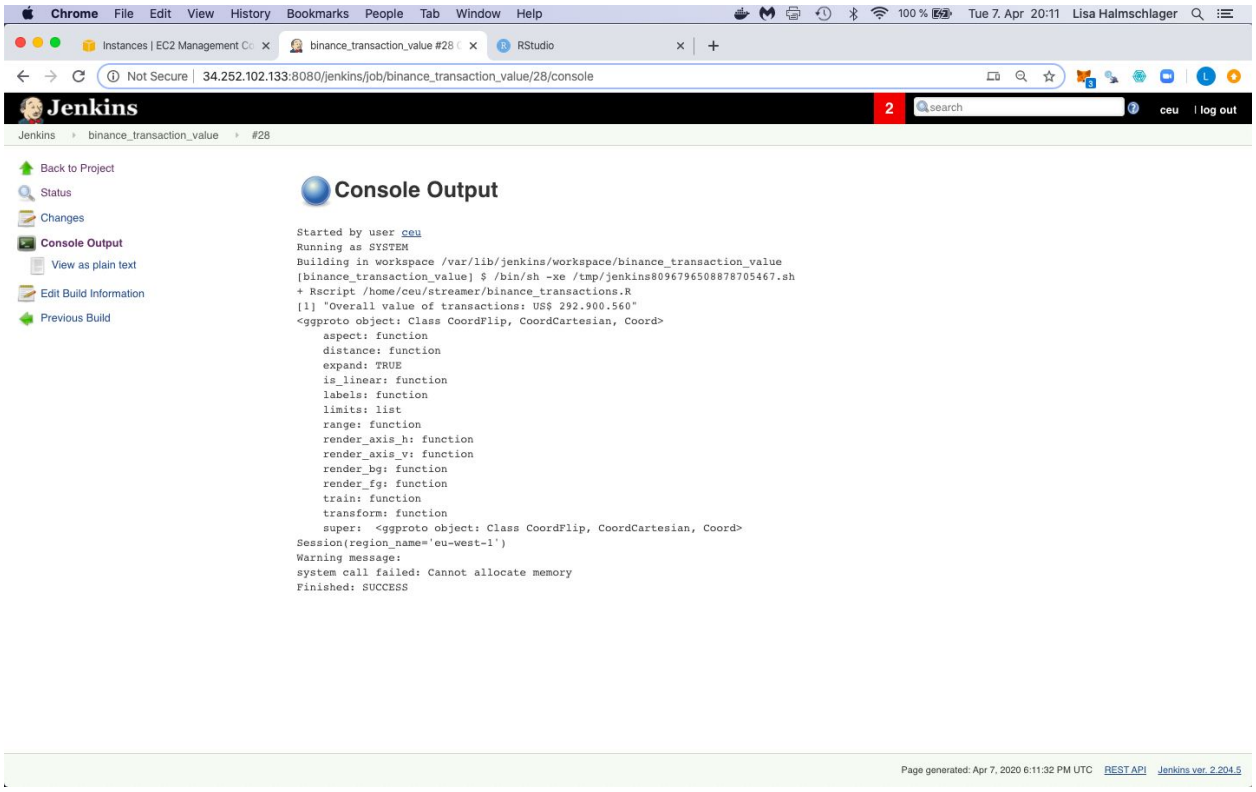
Jenkins job config page

- I logged in to **Jenkins** http://my.ip.address:8080/jenkins and the given credentials.
- I created a Jenkins job to check on the transactions following these steps:
    - New item
    - Enter name of job
    - Pick freestyle project
    - Add “Build periodically”: * * * * * for execution every minute
    - Add build step “Execute shell”
    - Enter command: `Rscript /home/ceu/streamer/binance_transactions.R`
    - Run the job (“Build now”)

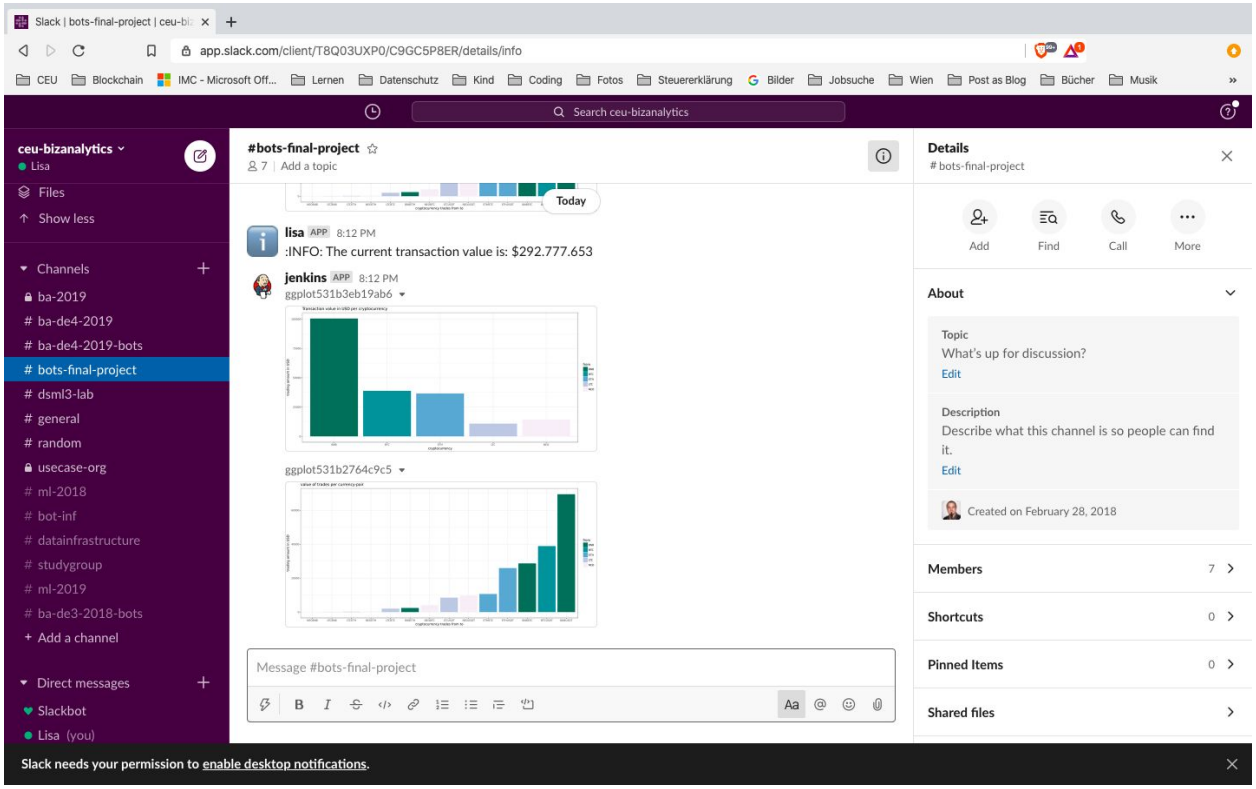




Jenkins job output



Slack channel notifications



Cleaning

Make sure to clean-up your EC2 nodes, security groups, keys etc created in the past weeks

- I stopped my instance(s)
- I deleted my unused security groups
- I deleted my unused key pairs

GLOSSARY

Amazon AWS stands for Amazon Web Services that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.

Amazon Machine Image (AMI) is a special type of virtual appliance. It is used to create a virtual machine within the Amazon Elastic Compute Cloud. The main component of an AMI is a read-only filesystem image that includes an operating system (e.g., Linux, Unix, or Windows) and any additional software required to deliver a service or a portion of it.

AWS EC2 or Amazon Elastic Compute Cloud provides scalable computing capacity in the Amazon Web Services (AWS) cloud. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage.

AWS IAM role is used in AWS Identity and Access Management. You can use roles to manage permissions, for example delegate access to users, applications, or services that don't normally have access to your AWS resources.

AWS Java SDK (software development kit) provides a Java API for Amazon Web Services. Using the SDK, you can easily build Java applications that work with Amazon S3, Amazon EC2, Amazon SimpleDB, and more.

AWS Key Management Service (KMS) is an Amazon Web Services product that allows administrators to create, delete and control keys that encrypt data stored in AWS databases and products. AWS KMS can be accessed within AWS Identity and Access Management by selecting the "Encryption Keys" section or by using the AWS KMS command-line interface or software development kit.

AWS Security group is associated with an EC2 instances and provides security at the protocol and port access level. Each security group — working much the same way as a firewall — contains a set of rules that filter traffic coming into and out of an EC2 instance.

CEU-Binance stream provides access to the real-time order data from the Binance cryptocurrency exchange on Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), Neo (NEO), Binance Coin (BNB) and Tether (USDT) -- including the attributes of each transaction as specified at https://github.com/binance-exchange/binance-official-api-docs/blob/master/web-socket-streams.md#trade-streams

Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

Redis is an in-memory data structure project implementing a distributed, in-memory key-value database with optional durability. It is used as a database, cache and message broker.

Slack is a proprietary instant messaging platform.