

+++ Step 1.2.4 Paste the following text into Amazon Kinesis Analytics application editor

-- Create intermediate Stream for incoming data

CREATE STREAM "INTERMEDIATE\_STREAM" (

hostname VARCHAR(1024),

logname VARCHAR(1024),

username VARCHAR(1024),

requesttime VARCHAR(1024),

request VARCHAR(1024),

status VARCHAR(32),

responsesize VARCHAR(32)

);

-- Data Pump: Take incoming data from SOURCE\_SQL\_STREAM\_001 and insert into INTERMEDIATE\_STREAM

CREATE OR REPLACE PUMP "INTERMEDIATE\_STREAM\_PUMP" AS

INSERT INTO "INTERMEDIATE\_STREAM"

SELECT STREAM

l.r.COLUMN1,

l.r.COLUMN2,

l.r.COLUMN3,

l.r.COLUMN4,

l.r.COLUMN5,

l.r.COLUMN6,

l.r.COLUMN7

FROM (SELECT STREAM W3C\_LOG\_PARSE("browseraction", 'COMMON')

FROM "SOURCE\_SQL\_STREAM\_001"

) AS l(r);

-- Define output stream that stores request counts by status

CREATE OR REPLACE STREAM "DESTINATION\_SQL\_STREAM" (

status VARCHAR(32),

requestCount INTEGER);

-- Data Pump: Take INTERMEDIATE\_STREAM and group into 1-minute intervals

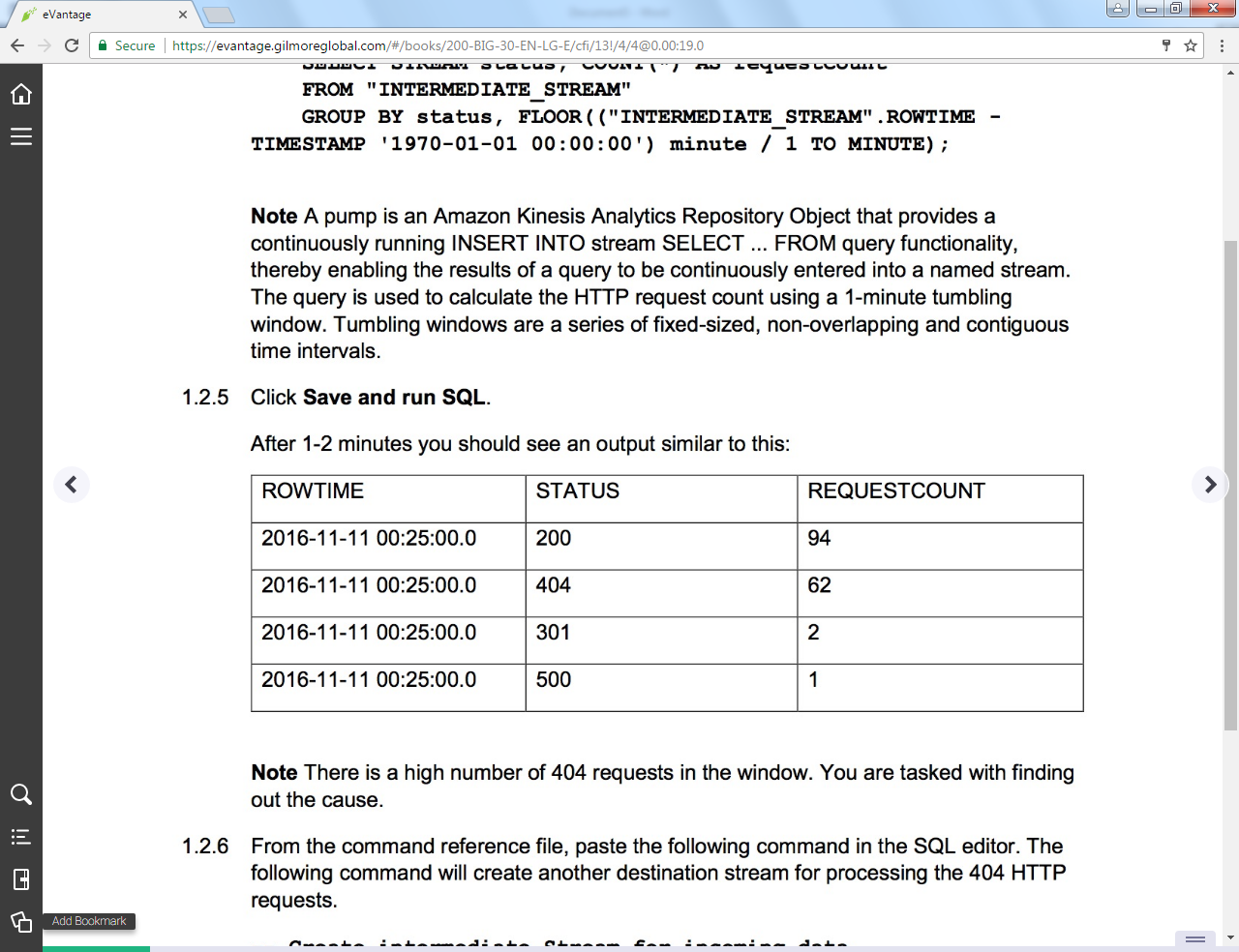
CREATE OR REPLACE PUMP "OUTPUT\_PUMP" AS

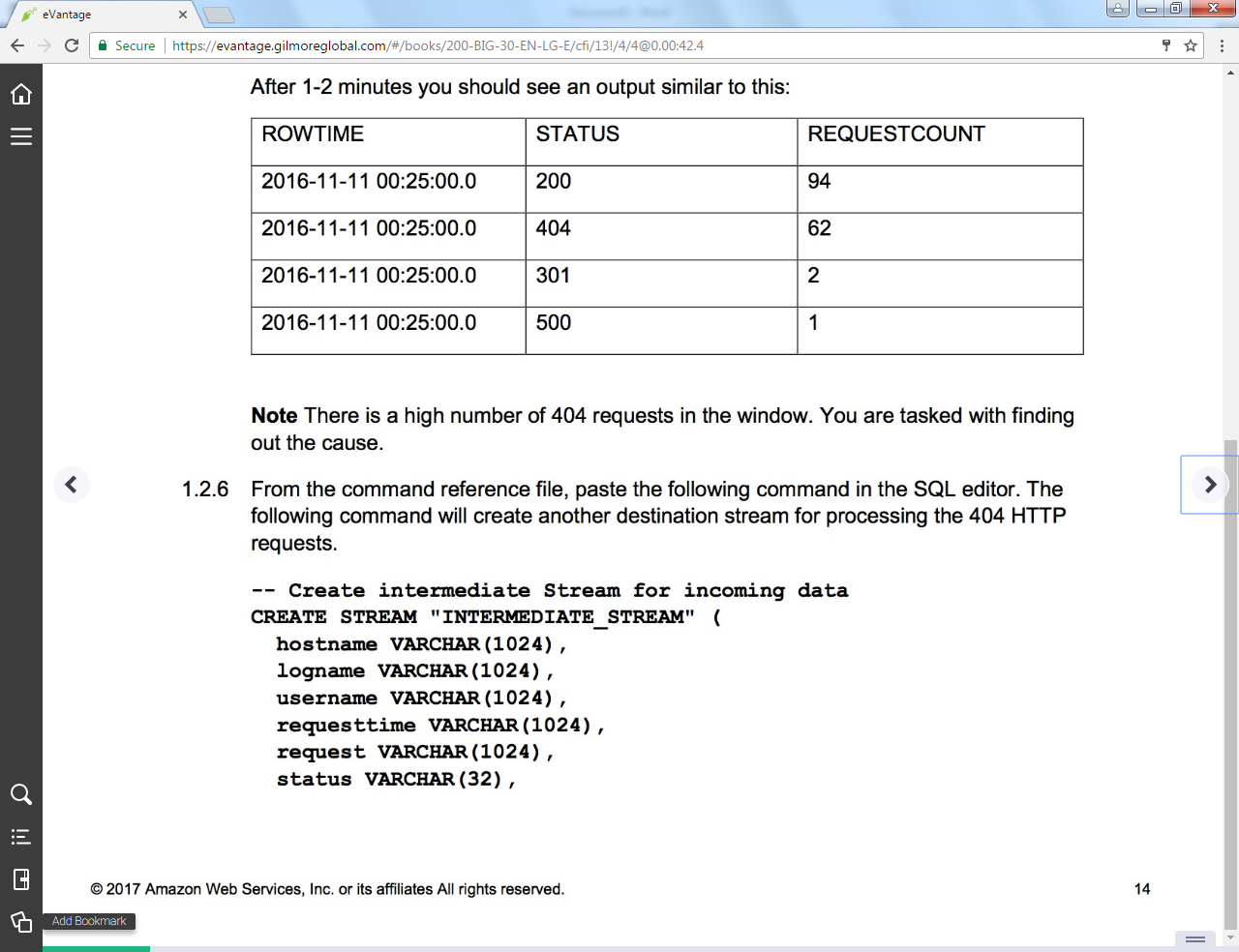
INSERT INTO "DESTINATION\_SQL\_STREAM"

SELECT STREAM status, COUNT(\*) AS requestCount

FROM "INTERMEDIATE\_STREAM"

GROUP BY status, FLOOR(("INTERMEDIATE\_STREAM".ROWTIME - TIMESTAMP '1970-01-01 00:00:00') minute / 1 TO MINUTE);





+++ Step 1.2.6 Paste text into Amazon Kinesis Analytics application editor

-- Create intermediate Stream for incoming data

CREATE STREAM "INTERMEDIATE\_STREAM" (

hostname VARCHAR(1024),

logname VARCHAR(1024),

username VARCHAR(1024),

requesttime VARCHAR(1024),

request VARCHAR(1024),

status VARCHAR(32),

responsesize VARCHAR(32)

);

-- Data Pump: Take incoming data from SOURCE\_SQL\_STREAM\_001 and insert into INTERMEDIATE\_STREAM

CREATE OR REPLACE PUMP "INTERMEDIATE\_STREAM\_PUMP" AS

INSERT INTO "INTERMEDIATE\_STREAM"

SELECT STREAM

l.r.COLUMN1,

l.r.COLUMN2,

l.r.COLUMN3,

l.r.COLUMN4,

l.r.COLUMN5,

l.r.COLUMN6,

l.r.COLUMN7

FROM (SELECT STREAM W3C\_LOG\_PARSE("browseraction", 'COMMON')

FROM "SOURCE\_SQL\_STREAM\_001"

) AS l(r);

-- Define an output stream that stores request counts by status

CREATE OR REPLACE STREAM "DESTINATION\_SQL\_STREAM" (

status VARCHAR(32),

requestCount INTEGER);

-- Data Pump: Take INTERMEDIATE\_STREAM and group into 1-minute intervals

CREATE OR REPLACE PUMP "OUTPUT\_PUMP" AS

INSERT INTO "DESTINATION\_SQL\_STREAM"

SELECT STREAM status, COUNT(\*) AS requestCount

FROM "INTERMEDIATE\_STREAM"

GROUP BY status, FLOOR(("INTERMEDIATE\_STREAM".ROWTIME - TIMESTAMP '1970-01-01 00:00:00') minute / 1 TO MINUTE);

-- Define an output stream that stores HTTP request and status

CREATE OR REPLACE STREAM "DESTINATION\_SQL\_STREAM\_404" (

status VARCHAR(32),

request VARCHAR(64));

-- Data Pump: Store status and requests in DESTINATION\_SQL\_STREAM\_404 stream

CREATE OR REPLACE PUMP "OUTPUT\_PUMP\_404" AS

INSERT INTO "DESTINATION\_SQL\_STREAM\_404"

SELECT STREAM status,request

FROM "INTERMEDIATE\_STREAM"

WHERE "INTERMEDIATE\_STREAM"."STATUS" = '404';

