

Created two S3 buckets

1.mybucket1-source

2.mybucket1-destination

Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

Total storage: Pending | Object count: Pending | Average object size: Pending

You can enable advanced metrics in the "default-account-dashboard" configuration.

General purpose buckets (3)

Buckets are containers for data stored in S3.

[Find buckets by name](#)

Name	AWS Region	Access	Creation date
garry11	US East (N. Virginia) us-east-1	Public	March 7, 2024, 17:44:00 (UTC+05:30)
mybucket1-destination	US East (N. Virginia) us-east-1	Bucket and objects not public	March 9, 2024, 11:56:46 (UTC+05:30)
mybucket1-source	US East (N. Virginia) us-east-1	Bucket and objects not public	March 9, 2024, 11:56:15 (UTC+05:30)

Created two folders in source bucket

mybucket1-source

[Objects](#) | [Properties](#) | [Permissions](#) | [Metrics](#) | [Management](#) | [Access Points](#)

Objects (2)

[Copy S3 URI](#) | [Copy URL](#) | [Download](#) | [Open](#) | [Delete](#) | [Actions](#) | [Create folder](#)

[Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Find objects by prefix](#)

Name	Type	Last modified	Size	Storage class
excel_file/	Folder	-	-	-
zip_file/	Folder	-	-	-

Created Lambda function on S3 bucket

The screenshot shows the AWS Lambda console for a function named 'myfunction_zip'. The 'Function overview' tab is active, displaying a diagram of the function's architecture. It shows an S3 bucket as the trigger, with a Lambda function 'myfunction_zip' connected to it. The function has one layer. The 'Code source' tab is also visible, showing the 'Upload from' button. The 'Description' section on the right provides details about the function, including its last modified time (1 hour ago), its ARN, and its URL.

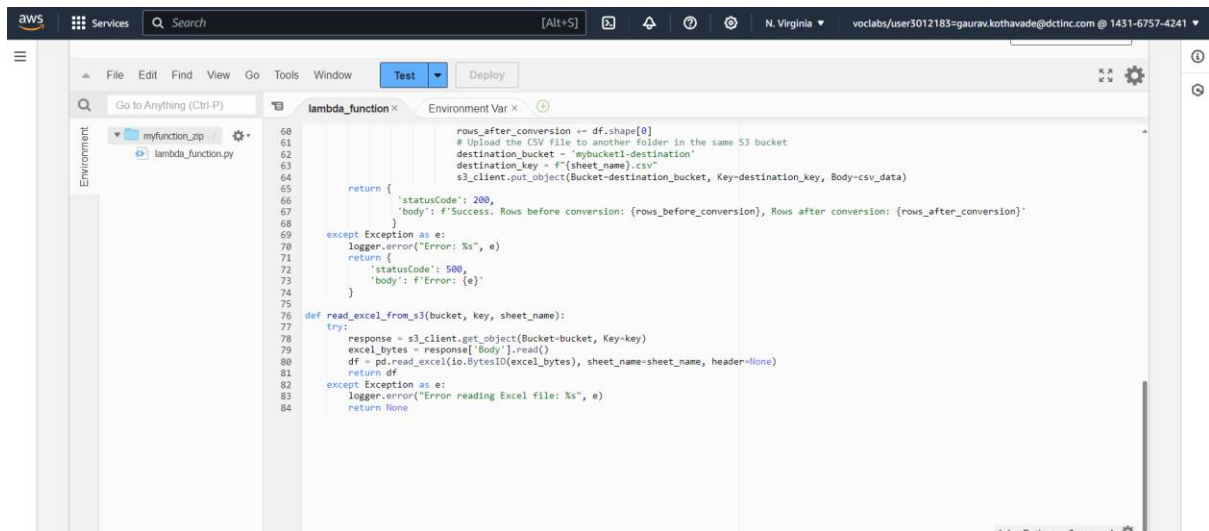
Code to trigger lambda

The screenshot shows the AWS Lambda console for the 'myfunction_zip' function, with the 'Code' tab selected. The code is written in Python and is triggered by an S3 event. The code defines a function 'default_columns' that takes a DataFrame and a file name as input. It then defines a 'lambda_handler' function that takes an event and a context as input. The handler extracts the S3 bucket and key from the event, downloads the file from the bucket, and processes it. The code is as follows:

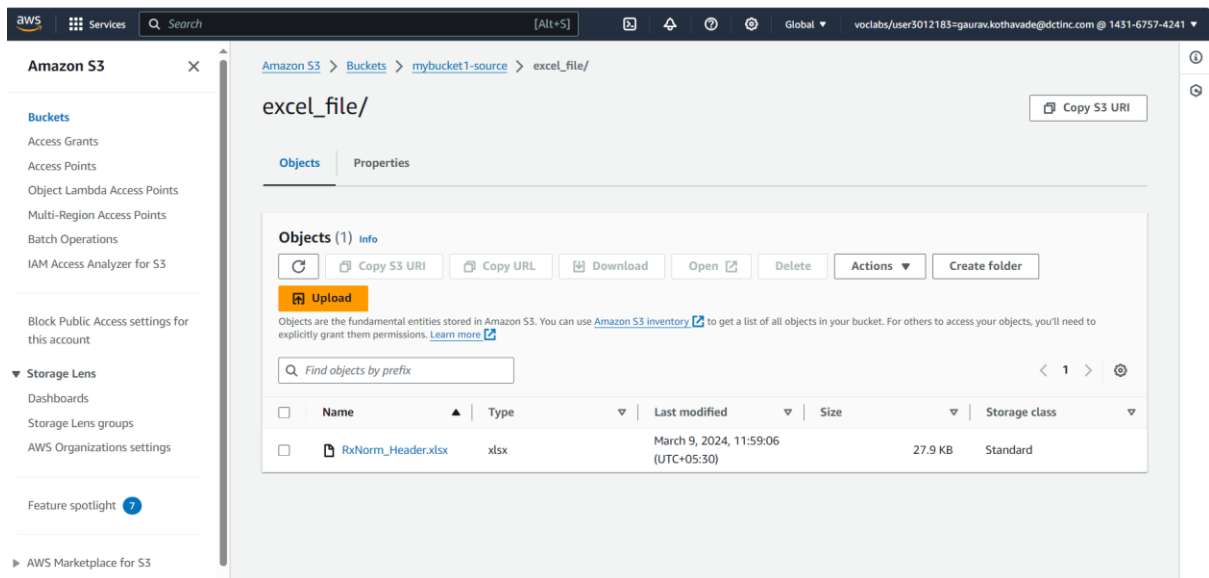
```
1 import os
2 import boto3
3 import zipfile
4 import io
5 import logging
6 from datetime import datetime
7 import pandas as pd
8
9 logger = logging.getLogger()
10 logger.setLevel(logging.INFO)
11
12 s3_client = boto3.client('s3')
13
14
15 def default_columns(df, file_name):
16     if 'RXNATOMARCHIVE' in file_name:
17         df['CREATED_TIMESTAMP'] = df['CREATED_TIMESTAMP'].apply(lambda x: datetime.strptime(x, "%m/%d/%Y %I:%M:%S %p").strftime("%Y-%m-%d") if pd.notnull(x) else '')
18         df['UPDATED_TIMESTAMP'] = df['UPDATED_TIMESTAMP'].apply(lambda x: datetime.strptime(x, "%m/%d/%Y %I:%M:%S %p").strftime("%Y-%m-%d") if pd.notnull(x) else '')
19         df['LAST_RELEASED'] = df['LAST_RELEASED'].fillna('').astype(str).apply(lambda x: datetime.strptime(x, "%d-%b-%y").strftime("%Y-%m-%d") if pd.notnull(x) else '')
20     elif 'RXNOSAB' in file_name:
21         df['VSTART'] = df['VSTART'].apply(lambda x: datetime.strptime(x, "%Y-%m-%d").strftime("%Y-%m-%d") if pd.notnull(x) and x != '' else x)
22         df['VEND'] = df['VEND'].apply(lambda x: datetime.strptime(x, "%Y-%m-%d").strftime("%Y-%m-%d") if pd.notnull(x) and x != '' else x)
23
24 def lambda_handler(event, context):
25     try:
26         logger.info("Received event: %s", event)
27         rows_before_conversion = 0
28         rows_after_conversion = 0
29
30         # Get the S3 bucket and object key from the event
31         bucket = event['Records'][0]['s3']['bucket']['name']
32         key = event['Records'][0]['s3']['object']['key']
33         RX_file_name = os.path.basename(key)
34
35         # Download the zip file from S3
36         response = s3_client.get_object(Bucket=bucket, Key=key)
37         zip_data = response['Body'].read()
38
39         # Extract the zip file
40         with zipfile.ZipFile(io.BytesIO(zip_data)) as zip_ref:
41             for file_name in zip_ref.namelist():
42                 if file_name.endswith(".RMP"):
43                     # Extract the file
44                     with zip_ref.open(file_name) as file_data:
45                         df = pd.read_csv(file_data, delimiter=';', low_memory = False)
46                         rows_before_conversion += len(df)
47                         # Get header names from the corresponding Excel sheet
48                         num_cols_df = df.shape[1]
49                         sheet_name = os.path.splitext(os.path.basename(file_name))[0]
50                         header_file_key = 'excel_file/RxNorm_Header.xlsx'
51                         header_df = read_excel_from_s3(bucket, header_file_key, sheet_name)
52                         if header_df is not None:
53                             df.columns = header_df.iloc[:, 0].tolist()[1:num_cols_df]
54                             default_columns(df, file_name)
55                         df['Code set'] = 'Rxnorm'
56                         version_month = datetime.strptime(RX_file_name.split("-")[1].split(".")[0], "%m-%d-%Y")
57                         df['Version Month'] = version_month.strftime("%Y-%m-%d")
58                         # Convert DataFrame to CSV format
59                         csv_data = df.to_csv(index=False)
60                         rows_after_conversion += df.shape[0]
61                         # Upload the CSV file to another folder in the same S3 bucket
62                         destination_bucket = 'mybucketl-destination'
63                         destination_key = f'{sheet_name}.csv'
64                         s3_client.put_object(Bucket=destination_bucket, Key=destination_key, Body=csv_data)
```

The screenshot shows the AWS Lambda console for the 'myfunction_zip' function, with the 'Code' tab selected. The code is written in Python and is triggered by an S3 event. The code defines a function 'default_columns' that takes a DataFrame and a file name as input. It then defines a 'lambda_handler' function that takes an event and a context as input. The handler extracts the S3 bucket and key from the event, downloads the file from the bucket, and processes it. The code is as follows:

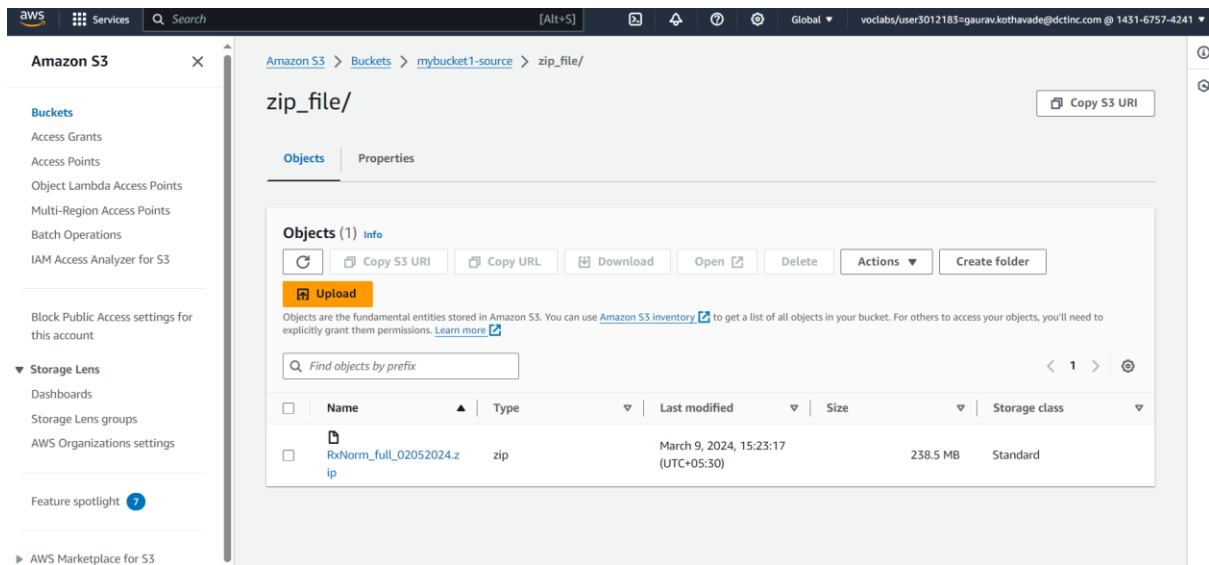
```
33 RX_file_name = os.path.basename(key)
34
35 # Download the zip file from S3
36 response = s3_client.get_object(Bucket=bucket, Key=key)
37 zip_data = response['Body'].read()
38
39 # Extract the zip file
40 with zipfile.ZipFile(io.BytesIO(zip_data)) as zip_ref:
41     for file_name in zip_ref.namelist():
42         if file_name.endswith(".RMP"):
43             # Extract the file
44             with zip_ref.open(file_name) as file_data:
45                 df = pd.read_csv(file_data, delimiter=';', low_memory = False)
46                 rows_before_conversion += len(df)
47                 # Get header names from the corresponding Excel sheet
48                 num_cols_df = df.shape[1]
49                 sheet_name = os.path.splitext(os.path.basename(file_name))[0]
50                 header_file_key = 'excel_file/RxNorm_Header.xlsx'
51                 header_df = read_excel_from_s3(bucket, header_file_key, sheet_name)
52                 if header_df is not None:
53                     df.columns = header_df.iloc[:, 0].tolist()[1:num_cols_df]
54                     default_columns(df, file_name)
55                 df['Code set'] = 'Rxnorm'
56                 version_month = datetime.strptime(RX_file_name.split("-")[1].split(".")[0], "%m-%d-%Y")
57                 df['Version Month'] = version_month.strftime("%Y-%m-%d")
58                 # Convert DataFrame to CSV format
59                 csv_data = df.to_csv(index=False)
60                 rows_after_conversion += df.shape[0]
61                 # Upload the CSV file to another folder in the same S3 bucket
62                 destination_bucket = 'mybucketl-destination'
63                 destination_key = f'{sheet_name}.csv'
64                 s3_client.put_object(Bucket=destination_bucket, Key=destination_key, Body=csv_data)
```



Upload Header excel file in source bucket



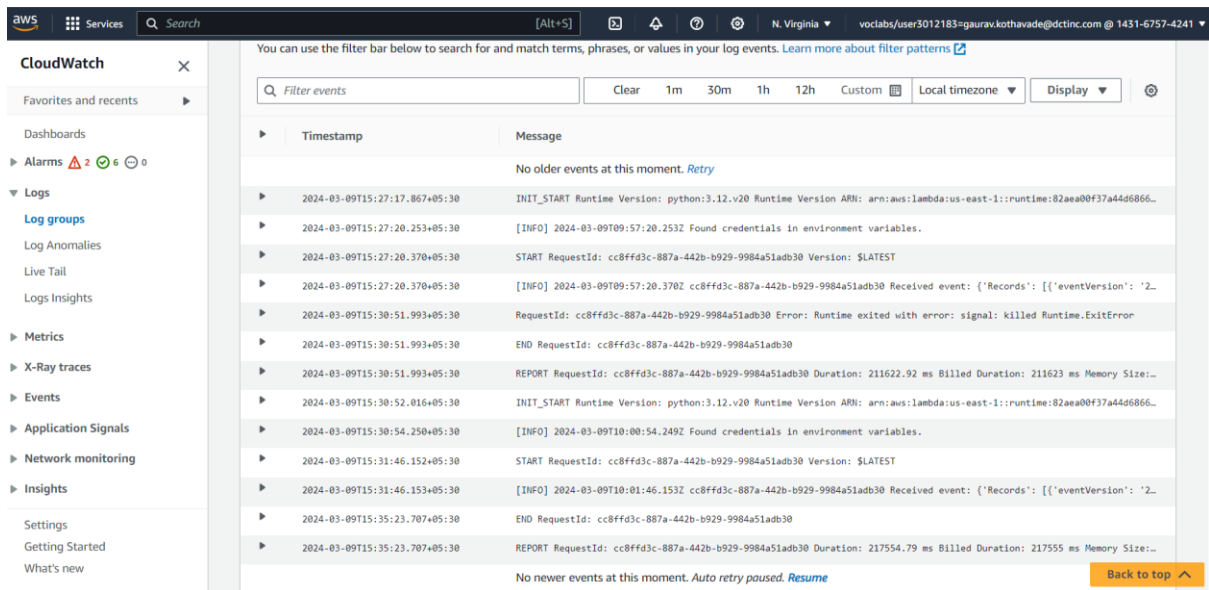
Upload the zip file in Source bucket to trigger lambda



Amazon S3 console view of the 'zip_file/' bucket. The 'Objects' tab is selected, showing a list of objects. The object 'RxNorm_full_02052024.zip' is visible, with a size of 238.5 MB and a storage class of 'Standard'. The 'Upload' button is highlighted in orange.

Name	Type	Last modified	Size	Storage class
RxNorm_full_02052024.zip	zip	March 9, 2024, 15:23:17 (UTC+05:30)	238.5 MB	Standard

Lambda triggered successfully after uploading .zip file



CloudWatch console view showing the logs for a Lambda function. The logs show the function execution starting at 2024-03-09T15:27:17.867+05:30, with messages indicating the runtime version, environment variables, and the start of the event processing. The logs end at 2024-03-09T15:35:23.707+05:30, showing the function completed successfully.

Timestamp	Message
2024-03-09T15:27:17.867+05:30	INIT_START Runtime Version: python:3.12.v20 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:82aea0f37a44d6866...
2024-03-09T15:27:20.253+05:30	[INFO] 2024-03-09T09:57:20.253Z Found credentials in environment variables.
2024-03-09T15:27:20.370+05:30	START RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30 Version: \$LATEST
2024-03-09T15:27:20.370+05:30	[INFO] 2024-03-09T09:57:20.370Z cc8ffd3c-887a-442b-b929-9984a51adb30 Received event: {'Records': [{'eventVersion': '2...
2024-03-09T15:30:51.993+05:30	RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30 Error: Runtime exited with error: signal: killed Runtime.ExitError
2024-03-09T15:30:51.993+05:30	END RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30
2024-03-09T15:30:51.993+05:30	REPORT RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30 Duration: 211622.92 ms Billed Duration: 211623 ms Memory Size:...
2024-03-09T15:30:52.016+05:30	INIT_START Runtime Version: python:3.12.v20 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:82aea0f37a44d6866...
2024-03-09T15:30:54.250+05:30	[INFO] 2024-03-09T10:00:54.249Z Found credentials in environment variables.
2024-03-09T15:31:46.152+05:30	START RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30 Version: \$LATEST
2024-03-09T15:31:46.153+05:30	[INFO] 2024-03-09T10:01:46.153Z cc8ffd3c-887a-442b-b929-9984a51adb30 Received event: {'Records': [{'eventVersion': '2...
2024-03-09T15:35:23.707+05:30	END RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30
2024-03-09T15:35:23.707+05:30	REPORT RequestId: cc8ffd3c-887a-442b-b929-9984a51adb30 Duration: 217554.79 ms Billed Duration: 217555 ms Memory Size:...

.rrf files got converted into .csv files along with headers and get stored into destination bucket

Amazon S3

Objects (9)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
RXNCUL.csv	csv	March 9, 2024, 15:33:12 (UTC+05:30)	2.1 MB	Standard
RXNCUICHANGES.csv	csv	March 9, 2024, 15:33:12 (UTC+05:30)	17.8 KB	Standard
RXNDOC.csv	csv	March 9, 2024, 15:33:12 (UTC+05:30)	271.7 KB	Standard
RXNREL.csv	csv	March 9, 2024, 15:34:16 (UTC+05:30)	645.2 MB	Standard
RXNSAB.csv	csv	March 9, 2024, 15:34:23 (UTC+05:30)	9.7 KB	Standard
RXNSAT.csv	csv	March 9, 2024, 15:35:15 (UTC+05:30)	622.4 MB	Standard
RXNSTY.csv	csv	March 9, 2024, 15:35:24 (UTC+05:30)	26.1 MB	Standard

Before Header .rrf file

RXNCUICHANGES - Notepad

File Edit Format View Help

```
|RXNORM||1653118|2672916|
|RXNORM||1653119|2672917|
|RXNORM||1806685|2672352|
|RXNORM||1806938|2673304|
|RXNORM||2647628|2672353|
1917446|4007247|VANDF|CD|LEVOCARNITINE 200MG/ML INJ|1791371|1791374|
2368358|1918|MMSL|BD|Isuprel HCl, 0.2 mg/mL intravenous solution|1667913|2673607|
2369900|2288|MMSL|BD|Methylene Blue, 10 mg/mL injectable solution|1737851|2672425|
2377629|753|MMSL|CD|carboprost 250 mcg/mL injectable solution|238014|2673110|
2645237|17256|GS|BD|Isuprel 1:5,000 Solution for Injection|1667913|2673607|
2646313|4177|GS|CD|Methylene Blue 1% Solution for injection|1737851|2673109|
3054488|35947|GS|CD|L-Carnitine 200mg/1mL Solution for injection|1791371|1791374|
3514018|26188|MMSL|BD|Hizentra, 20% subcutaneous solution|1809065|2673182|
3792519|126124|MMX|BD|Powder, Multi Ingredient Topical application Powder [ZEASORB]|1144113|1947925|
3793991|125914|MMX|CD|Risperidone 12.5 MG Intramuscular Powder for Suspension, Extended Release|1133452|706822|
3794003|123625|MMX|CD|Risperidone 37.5 MG Intramuscular Powder for Suspension, Extended Release|1131777|402011|
3794010|123626|MMX|CD|Risperidone 50 MG Intramuscular Powder for Suspension, Extended Release|1127829|402012|
3808194|1663|MMSL|BD|Levocarnitine, 200 mg/mL injectable solution|1791371|1791374|
5050200|0264-9872|MTHSPL|DP|HEPARIN SODIUM 200 [USP'U] in 100 mL INTRAVENOUS INJECTION|1658692|1658691|
5471392|004369|NDDF|IN|enalaprilat dihydrate|235964|2672730|
5480353|N06AX19|ATC|IN|gepirone|30080|2672253|
5941679|87006|GS|BD|Isuprel 1:5,000 Solution for Injection|1667913|2673607|
606755|12551|GS|BD|Isuprel 1:5,000 Solution for Injection|1667913|2673607|
```

After header .csv file

RXAUI													
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	RXAUI	CODE	SAB	TTY	STR	OLD_RXCUI	NEW_RXCUI	Code set	Version Month				
2			RXNORM			1653119	2672917	Rxnorm	05-02-2024				
3			RXNORM			1806685	2672352	Rxnorm	05-02-2024				
4			RXNORM			1806938	2673304	Rxnorm	05-02-2024				
5			RXNORM			2647628	2672353	Rxnorm	05-02-2024				
6	1917446	4007247	VANDF	CD	LEVOCARNITIN	1791371	1791374	Rxnorm	05-02-2024				
7	2368358	1918	MMSL	BD	Isuprel HCl, 0.2	1667913	2673607	Rxnorm	05-02-2024				
8	2369900	2288	MMSL	BD	Methylene Blue	1737851	2672425	Rxnorm	05-02-2024				
9	2377629	753	MMSL	CD	carboprost 250	238014	2673110	Rxnorm	05-02-2024				
10	2645237	17256	GS	BD	Isuprel 1:5,000	1667913	2673607	Rxnorm	05-02-2024				
11	2646313	4177	GS	CD	Methylene Blue	1737851	2673109	Rxnorm	05-02-2024				
12	3054488	35947	GS	CD	L-Carnitine 200	1791371	1791374	Rxnorm	05-02-2024				
13	3514018	26188	MMSL	BD	Hizentra, 20% s	1809065	2673182	Rxnorm	05-02-2024				
14	3792519	126124	MMX	BD	Powder, Multi Ir	1144113	1947925	Rxnorm	05-02-2024				
15	3793991	125914	MMX	CD	Risperidone 12	1133452	706822	Rxnorm	05-02-2024				
16	3794003	123625	MMX	CD	Risperidone 37	1131777	402011	Rxnorm	05-02-2024				
17	3794010	123626	MMX	CD	Risperidone 50	1127829	402012	Rxnorm	05-02-2024				
18	3808194	1663	MMSL	BD	LevOCARNitine	1791371	1791374	Rxnorm	05-02-2024				
19	5050200	0264-9872	MTHSPL	DP	HEPARIN SODIUM	1658692	1658691	Rxnorm	05-02-2024				
20	5471392	4369	NDDF	IN	enalaprilat dihy	235964	2672730	Rxnorm	05-02-2024				
21	5480353	N06AX19	ATC	IN	gepirone	30080	2672253	Rxnorm	05-02-2024				
22	5941679	87006	GS	BD	Isuprel 1:5,000	1667913	2673607	Rxnorm	05-02-2024				
23	6826765	43581	GS	CD	Paricalcitol 2m	1746490	1736931	Rxnorm	05-02-2024				
24	7243916	2672917	RXNORM	SY	chlophedianol H	1653119	2672917	Rxnorm	05-02-2024				
25	7275473	117339	MMX	CD	Insulin Glargine	2601843	311041	Rxnorm	05-02-2024				
26	7703797	130908	MMX	BD	Isoproterenol H	1667913	2673607	Rxnorm	05-02-2024				
27	7732614	1349	MMSL	BD	Bivigam, 10% ir	1809418	1809416	Rxnorm	05-02-2024				
28	8214229	132591	MMX	CD	Methylene Blue	1788990	1788984	Rxnorm	05-02-2024				
29	8224604	61953-0001	MTHSPL	DP	Albumin Human	1741302	1741304	Rxnorm	05-02-2024				