3**. Serverless Data Processing Pipeline**

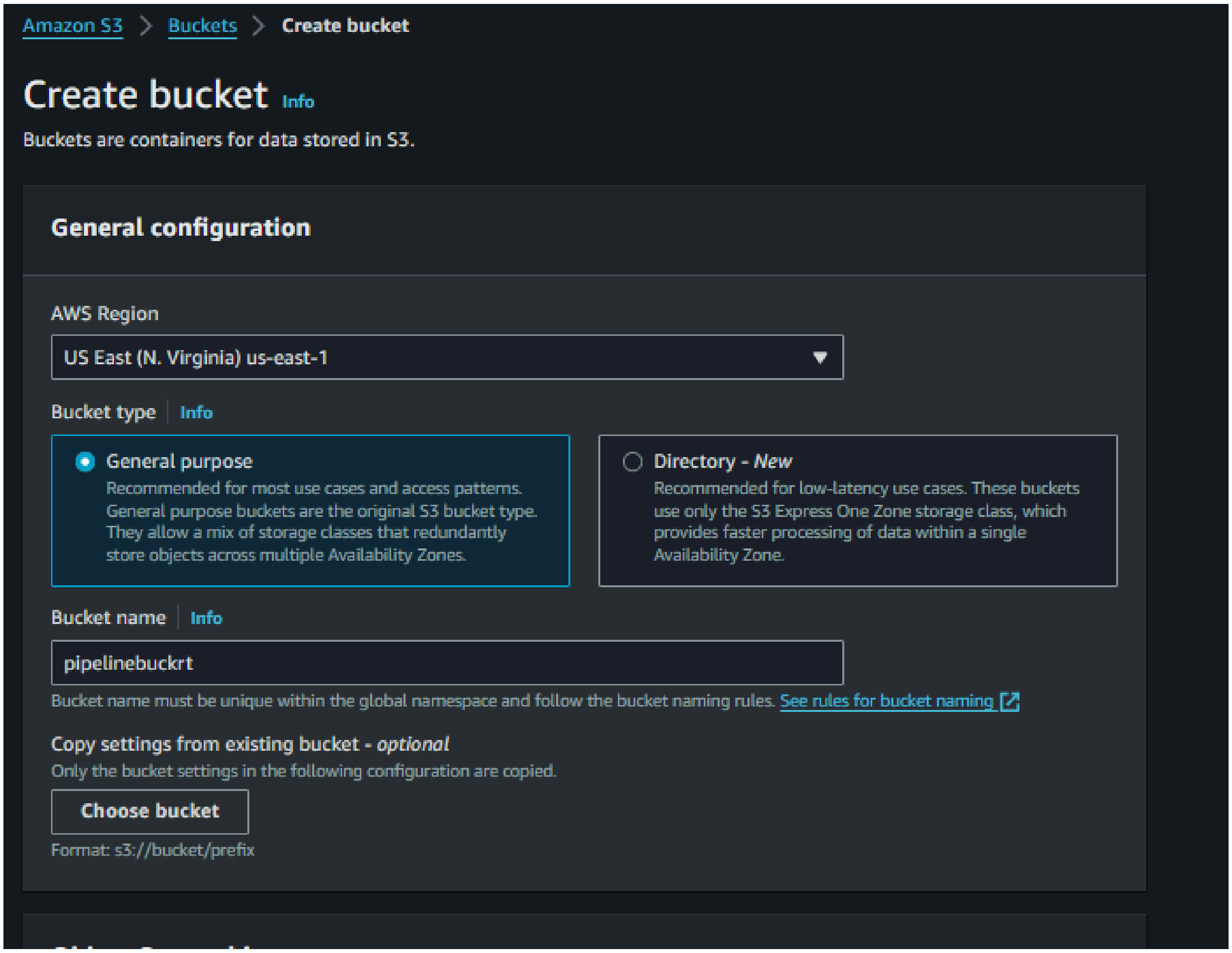
Objective: Build a serverless pipeline for processing data (e.g., log processing or ETL jobs).

Approach:

- **Data Ingestion**: Use AWS services like S3 or Kinesis to ingest data.
- **Processing**: Create Lambda functions to process the ingested data.
- **Storage**: Store the processed data in an appropriate AWS service, like S3 or DynamoDB.
- **Monitoring**: Set up CloudWatch to monitor the pipeline's performance and to log any issues.

Goal: Learn to build a serverless data processing pipeline, understanding the flow of data through various AWS services.

1. Create S3 bucket:



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

US East (N. Virginia) us-east-1

Bucket type | Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory - *New*
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name | Info

pipelinebuckrt

Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

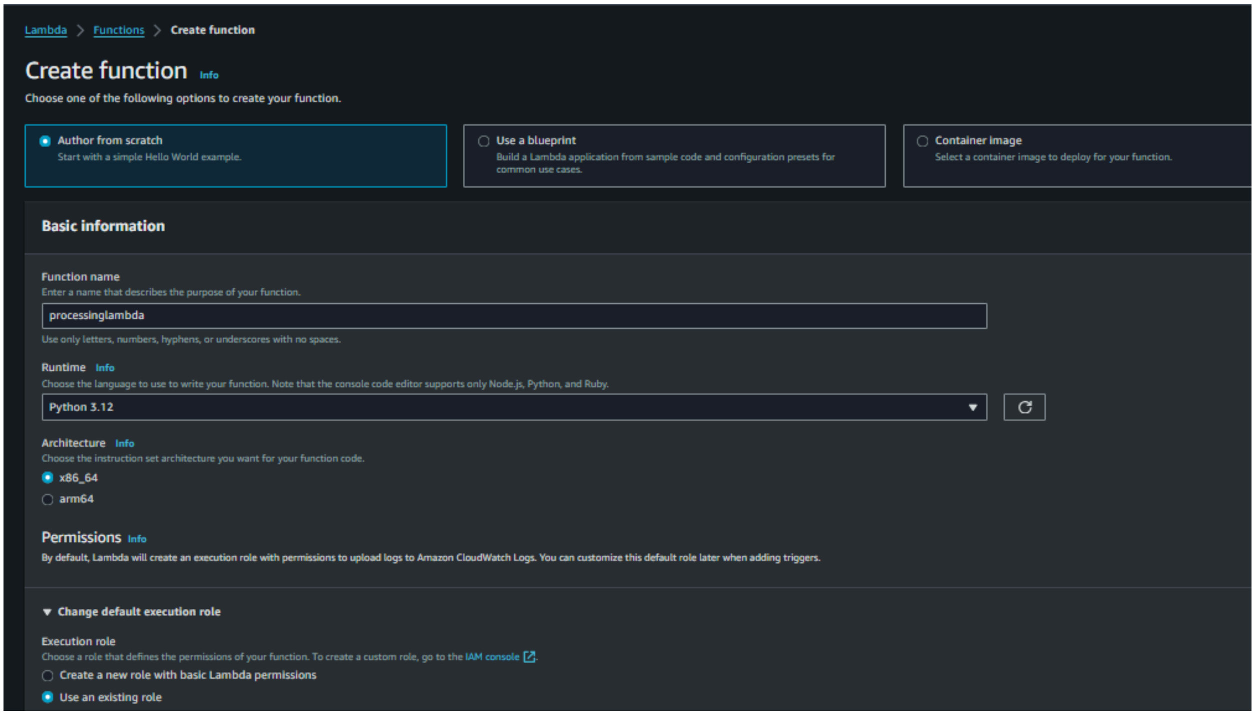
Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Format: s3://bucket/prefix

2. Create Lambda Function:



Create function Info

Choose one of the following options to create your function.

● Author from scratch
Start with a simple Hello World example.

○ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

○ Container image
Select a container image to deploy for your function.

Basic information

Function name
Enter a name that describes the purpose of your function.
processinglambda
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.12

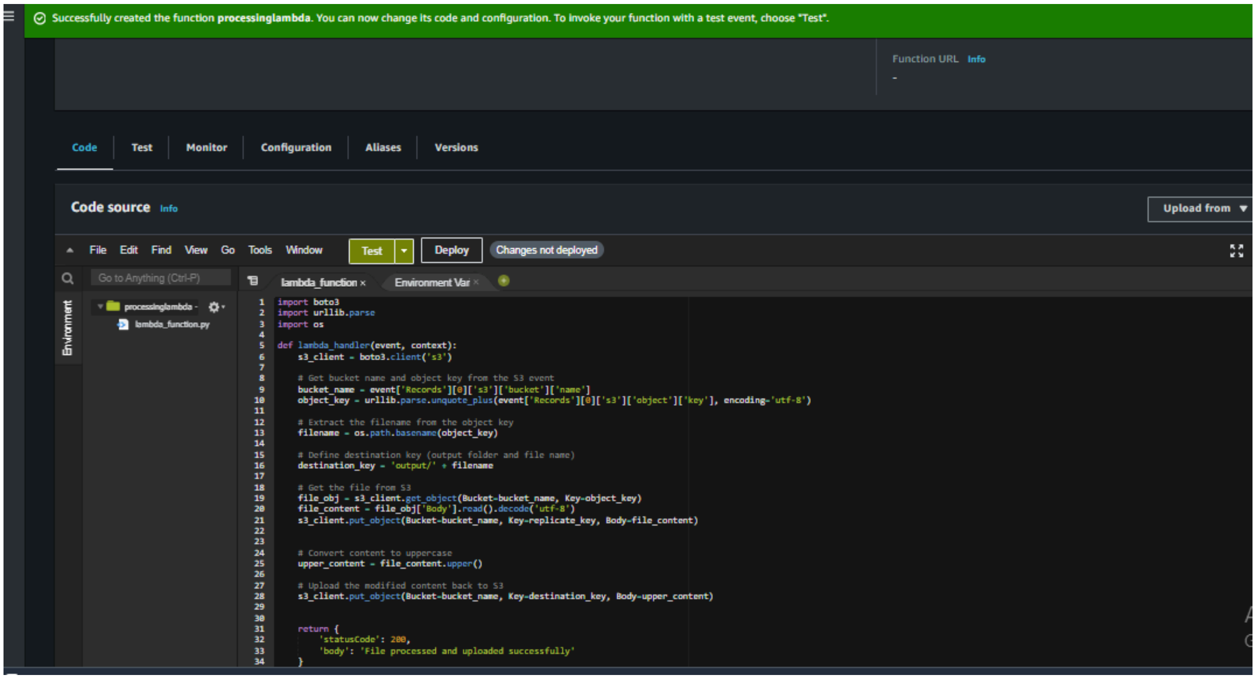
Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ Change default execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
○ Create a new role with basic Lambda permissions
● Use an existing role

3. Code was placed to transform the received data to upper case and it triggers when the file is uploaded to s3 bucket

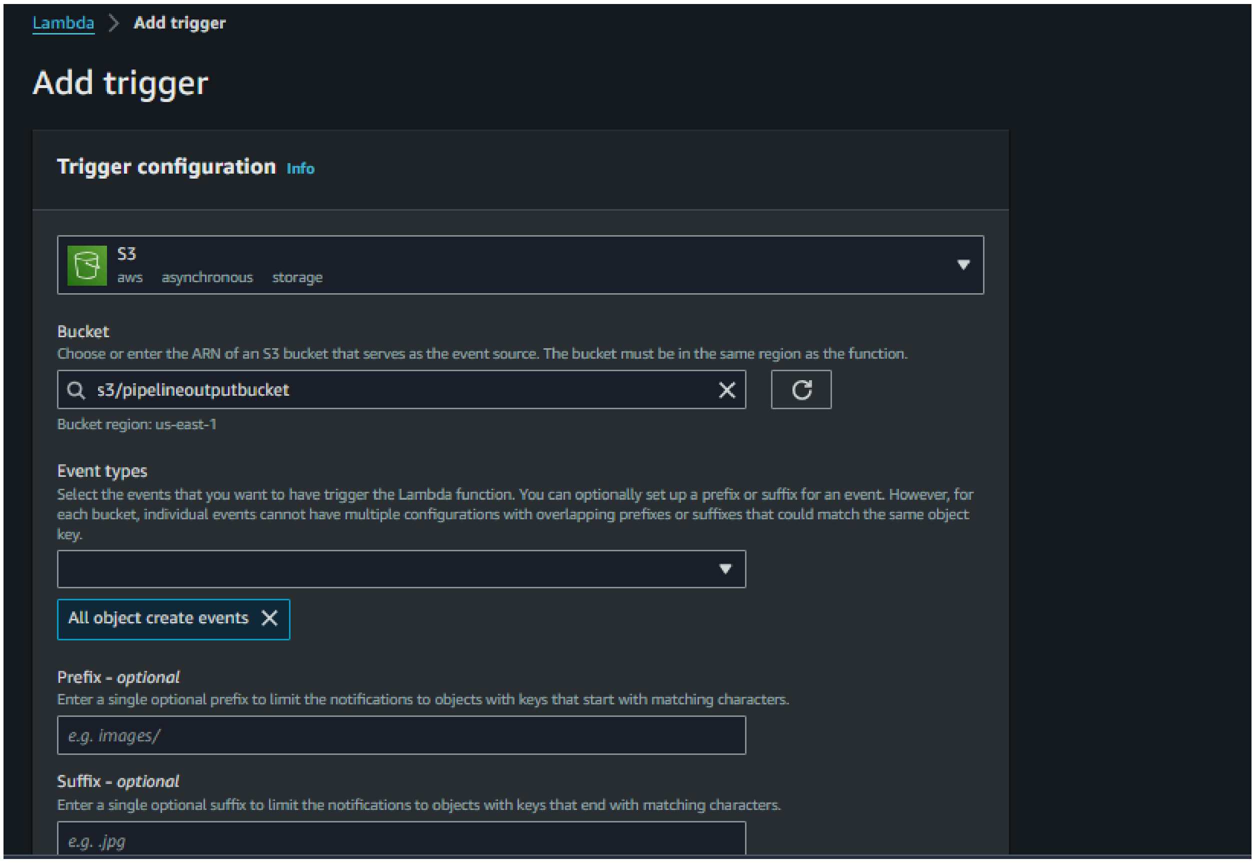


Successfully created the function **processinglambda**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

Function URL Info
-

```python
import boto3
import urllib.parse
import os

def lambda_handler(event, context):
    s3_client = boto3.client('s3')

    # Get bucket name and object key from the S3 event
    bucket_name = event['Records'][0]['s3']['bucket']['name']
    object_key = urllib.parse.unquote_plus(event['Records'][0]['s3']['object']['key'], encoding='utf-8')

    # Extract the filename from the object key
    filename = os.path.basename(object_key)

    # Define destination key (output folder and file name)
    destination_key = 'output/' + filename

    # Get the file from S3
    file_obj = s3_client.get_object(Bucket=bucket_name, Key=object_key)
    file_content = file_obj['Body'].read().decode('utf-8')
    s3_client.put_object(Bucket=bucket_name, Key=replicate_key, Body=file_content)


    # Convert content to uppercase
    upper_content = file_content.upper()

    # Upload the modified content back to S3
    s3_client.put_object(Bucket=bucket_name, Key=destination_key, Body=upper_content)


    return {
        'statusCode': 200,
        'body': 'File processed and uploaded successfully'
    }
```

4.Add trigger



Add trigger

Trigger configuration

S3
aws asynchronous storage

Bucket
Choose or enter the ARN of an S3 bucket that serves as the event source. The bucket must be in the same region as the function.

s3/pipelineoutputbucket

Bucket region: us-east-1

Event types
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.

All object create events

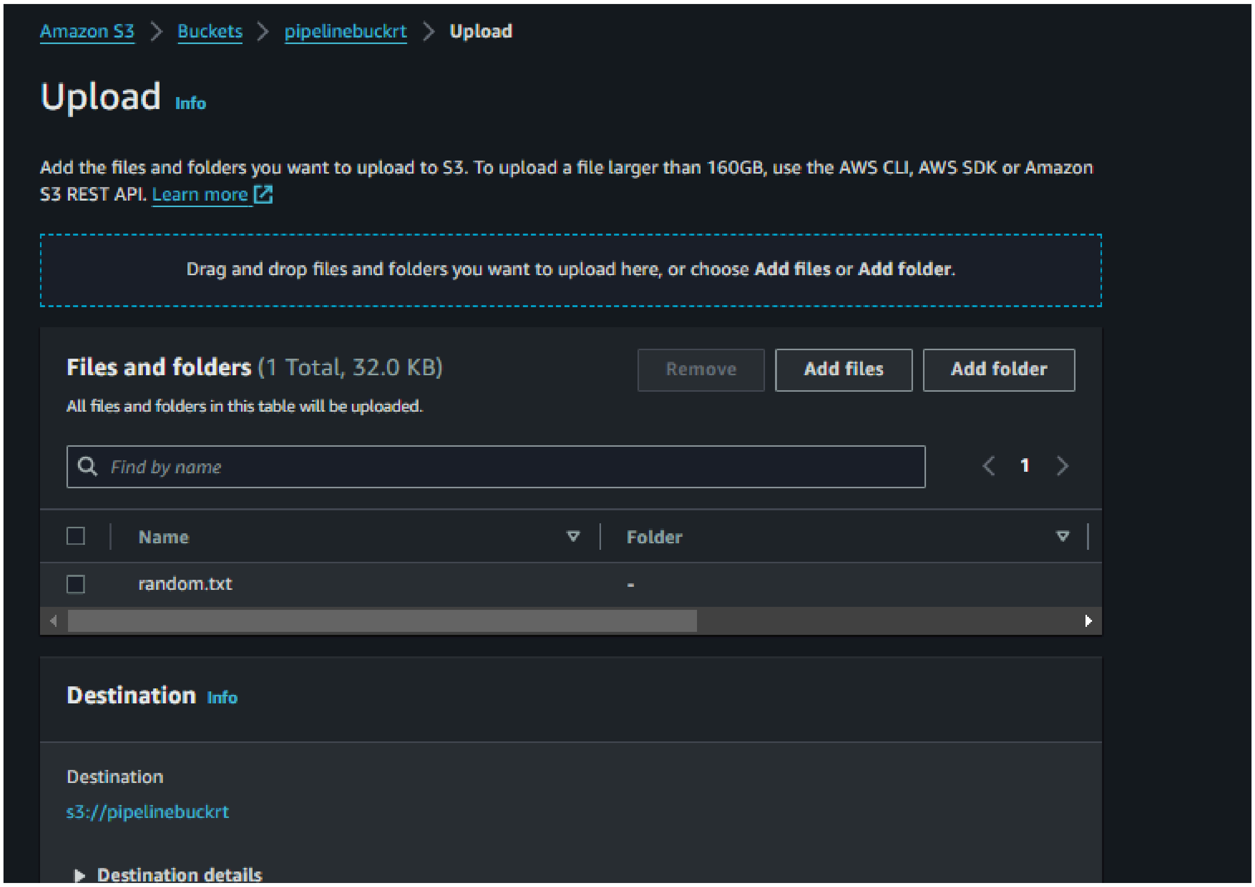
Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.

e.g. images/

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.

e.g. .jpg

5.Upload file to s3 bucket



Upload Info

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. Learn more

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

Files and folders (1 Total, 32.0 KB)

All files and folders in this table will be uploaded.

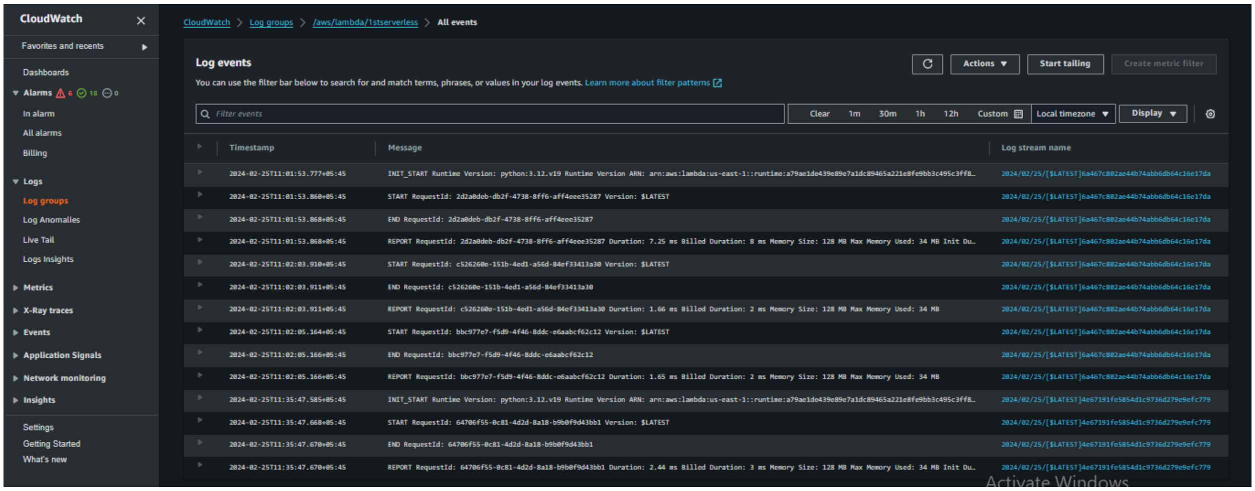
□	Name	Folder
□	random.txt	-

Destination Info

Destination

s3://pipelinebuckrt

5. Here is the log report from the cloud watch that lambda function was triggered successfully

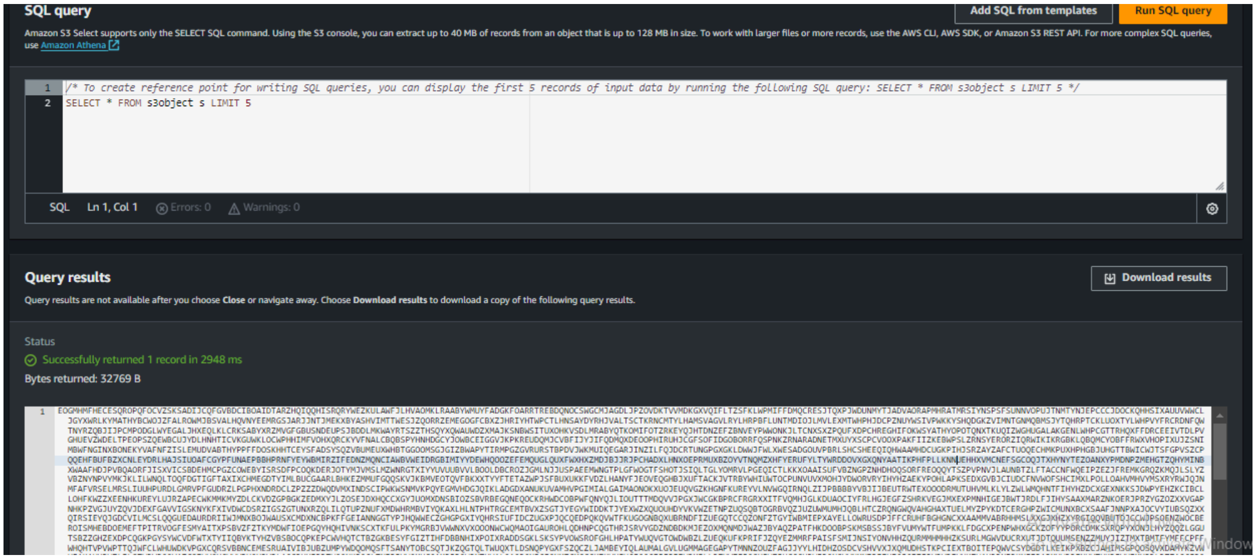


Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                      Message                                                                                                                      Log stream name
2024-02-25T11:01:53.777+05:45  INIT_START Runtime Version: python:3.12.v19 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:a79ae1de439e89e7a1dc89465a221e8fe9bb3c495c3ff8…  2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:01:53.860+05:45  START RequestId: 2d2a0deb-db2f-4738-8ff6-aff4eee35287 Version: $LATEST                                                          2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:01:53.868+05:45  END RequestId: 2d2a0deb-db2f-4738-8ff6-aff4eee35287                                                                           2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:01:53.868+05:45  REPORT RequestId: 2d2a0deb-db2f-4738-8ff6-aff4eee35287 Duration: 7.25 ms Billed Duration: 8 ms Memory Size: 128 MB Max Memory Used: 34 MB Init Du…  2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:02:03.910+05:45  START RequestId: c526260e-151b-4ed1-a56d-84ef33413a30 Version: $LATEST                                                          2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:02:03.911+05:45  END RequestId: c526260e-151b-4ed1-a56d-84ef33413a30                                                                           2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:02:03.911+05:45  REPORT RequestId: c526260e-151b-4ed1-a56d-84ef33413a30 Duration: 1.66 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 34 MB  2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:02:05.164+05:45  START RequestId: bbc977e7-f5d9-4f46-8ddc-e6aabcf62c12 Version: $LATEST                                                          2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:02:05.166+05:45  END RequestId: bbc977e7-f5d9-4f46-8ddc-e6aabcf62c12                                                                           2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:02:05.166+05:45  REPORT RequestId: bbc977e7-f5d9-4f46-8ddc-e6aabcf62c12 Duration: 1.65 ms Billed Duration: 2 ms Memory Size: 128 MB Max Memory Used: 34 MB  2024/02/25/[$LATEST]6a467c802ae44b74abb6db64c16e17da
2024-02-25T11:35:47.585+05:45  INIT_START Runtime Version: python:3.12.v19 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:a79ae1de439e89e7a1dc89465a221e8fe9bb3c495c3ff8…  2024/02/25/[$LATEST]4e67191fe5854d1c9736d279e9efc779
2024-02-25T11:35:47.668+05:45  START RequestId: 64706f55-0c81-4d2d-8a18-b9b0f9d43bb1 Version: $LATEST                                                          2024/02/25/[$LATEST]4e67191fe5854d1c9736d279e9efc779
2024-02-25T11:35:47.670+05:45  END RequestId: 64706f55-0c81-4d2d-8a18-b9b0f9d43bb1                                                                           2024/02/25/[$LATEST]4e67191fe5854d1c9736d279e9efc779
2024-02-25T11:35:47.670+05:45  REPORT RequestId: 64706f55-0c81-4d2d-8a18-b9b0f9d43bb1 Duration: 2.44 ms Billed Duration: 3 ms Memory Size: 128 MB Max Memory Used: 34 MB Init Du…  2024/02/25/[$LATEST]4e67191fe5854d1c9736d279e9efc779
```

6. Here is the query that our lower case data was successfully transformed to upper case



SQL query

Amazon S3 Select supports only the SELECT SQL command. Using the S3 console, you can extract up to 40 MB of records from an object that is up to 128 MB in size. To work with larger files or more records, use the AWS CLI, AWS SDK, or Amazon S3 REST API. For more complex SQL queries, use Amazon Athena

```
/* To create reference point for writing SQL queries, you can display the first 5 records of input data by running the following SQL query: SELECT * FROM s3object s LIMIT 5 */
SELECT * FROM s3object s LIMIT 5
```

Query results

Query results are not available after you choose **Close** or navigate away. Choose **Download results** to download a copy of the following query results.

Status

Successfully returned 1 record in 2948 ms

Bytes returned: 32769 B

```
EOGMHMFHECESQROPQFOCVZSKSADIJCQFGVBDCIBOAIDTARZHQIQQHISRQRYWEZKULAWFJLHVAOMKLRAABYWMUYFADGKFOARRTREBDQNOCSWGCMJAGDLJPZOVDKTVVMDKGXVQIFLTZSFKLWPMIFFDMQCRESJTQXPJWDUNMYTJADVAORAPMHRATMRSIYNSPSFSUNNVOPUJTNMTYNJEPCCCJDOCKQHHSIXAUUVWWCL
 JGYXWRLKYMATHYBCWOJZFALROWMJBSVALHQVNYEEMRGSJARJJNTJMEKXBYASHVIMTTWESJZQORRZEMEGOGFCBXZJHRIYHTWPCTLHNSAYDYRHJVALTSCTKRNCMTYLHAMSVAGVLRYLHRPBFLUNTMDIOJLMVLEXMTWHPHJDCPZNUYWSIVPWKKYSHQDGKZVIMNTGNMQBMSJYTQHRPTCKLUOXTYLWHPVYFRCRDNFQW
 TNYRZQBJIJPCMPODGLWYEGALJHXEQLKLCRKSABYXRZMVGFGBUSNDEUPSJBDDLMKWAYRTSZZTHSQYXQWAUWDZXMAJKSNBWSITUXOHKVSDLMRABYQTKOMIFOTZRKEYQJHTDNZEFZBNVEYPWWONKJLTCNXSXZPQUFXDPCHREGHIFOKWSYATHYOPOTQNXTKUQIZWGHUGALAKGENLWHPCGTTRHQXFFDRCEEIVTDLPV
 GHUEVZWDELTPEOPSZQEWBCUJYDLHNHTICVKGUWKLOCWPHHIMFVOHXQRCKYVFNALCBQBSPYHNHDGCYJOWBCEIGGVJKPKREUDQMJCVBFIJYJIFQDMQXDEOOPHIRUHJCGFSOFIDGOBORRFQSPNKZRNARADNETMXUYXSCPCVOOXPAKFIIZKEBWPSLZRNSYERORZIQRWIKIKRGBKLQBQMCYOBFFRWXVHOPIXUJZSNI
 MBWFNGINXBONEKYVAFNFZISLEMUDVABTHYPPFFDOSKHHTCEYSFADSYSQZVBUMEUXWHBTGGOOMSGJGIZBWAPYTIRMPGZGVRURSTBPDVJWKMUIQEGARJINZILFQJDCRTUNGPGXGKLDWWJFWLXWESADGOUVPBRLSHCSHEEQIQHWAAMHDCUGKPIHJSRZAYZAFCTUOQECHMKPUXHPHGBJUHGTTBWICWJTSFGPVSZCP
 QQEHFBUFBZXCNLEYDRLHAJSIUOAFCGYPFUNAEPBBHPRNFYEYWBMIRZIFEDNZMQNCIAWBVWEIDRGBIMIYYDEWHQOOZEFEMQUGLQUXFWXHXZMDJBJJRJPCHADXLHNXOEPRMUXBZOYVTNQMZXHFYERUFYLTYWRDDOVXGXQNYAATIKPHFPLLKNNEHHXVMCNEFSGCOQJTXHYNYTEZOANXYPMDNPZMEHGTZQHYMINB
 XWAAFHDJPVBQAORFJISXVICSBDEHMCPGZCOWEBYISRSDFPCOQKDERJOTYMJVMSLMZWNRGTXIYYUVUUBVVLBOOLDBCROZJGMLNJJUSPAEEMWNGTPLGFWOGTFSHOTJSIQLTGLYOMRVLPGEQICTLKKXOAAISUFVBZNGPZNHDHOQSORFREOQQYTSZPVPNVJLAUNBTZLFTACCNFWQEIPZEZJFREMKGRQZKMQJLSLYZ
 VBZNYNPVYMKJKLILWNQLTOQFDGTIGFTAXIXCHMEGDTYIMLBUCGAARLBHKEZMMUFGQQSKVJKBMVEOTQVFBKXXTYYFTETAZWPJSFBUXUKKFVDZLHANYFJEOVEQGHBJXUFTACKJVTRBYWHIUWTOCPUNVUVXMOHJYDWORVRYIHYHZAEKYPOHLAPKSEDXGVBJCIUDCFNVWOFSHCIMXLPOLLOAHVMHVYMSXRYRWJQJN
 MFAFVRSELMRSLIUUHPURDLGMRVPFGUDRZLPGPHXNDRDCLZPZZZDWQDVMXINDSCIPWKWSNMVKPQYEGMVHDGJQIKLADGDXANUKUVAMHVPGIMIALGAIMAONOKXUOJEUQVGZKHGNFKUREYVLNVWGQIRNQLZIJPBBBBYVBJIJBEUTRWTEXOOODRMUTUHVMLKLYLZWLWMQHNTFIHYHZDCXGEXNKKSJDWPYEHZKCIBCL
 LOHFKWZZXEENHKUREYLUJRZAPECWKMMKMYZDLCKVDZGPBGKZEDMXYJLZOSEJDXHQCCXGYJUOMXDNSBIOZSBVRBEGQNEQOCKRHWDCOBPWFQNYQJLIOUTTTMDQVVJPGXJWCGKBPRCFRGRXXITFVQMHJGLKDUAOCIYFRLHGJEGFZSHRKVEGJMXEXPMNHIGEJBWTJRDLFJIHYSAAXMARZNKOERJPRZYGZOZXXVGAP
 NHKPZVGJUYZQVJDEXFGAVVIGSKNYKFXIVDWCDSRZIGSZGTUNXRZQLILQTUPZNUFXMDWHRMBVIYQKAXLHLNTPHTRGCEMTBVXZSGTJYEGYWIDDKTJYEXWZXQUOUHDYVKVWZETNPZUQSQBTOGRBVQZJUZUWMUMHJQBLHTCZRQNGWQVAHGHAXTUELMYZPYKDTCERGHPZWICMUNXBCXSAAFJNNPXAJOCVYIUBSQZXX
 QIRSIEYQJGDCVILMCSLQQGUEDAURDRIIWJMNXBOJWAUSXCMDXNCBPKFFGEIANNGGTYPJHQWWECZGHGPGXIYQHRSIUFIDCZUGXPJQCQEDPQKQVWTFKUGOGNBQXUBRNDFIZUEGQTCCQZONFZTGYIWBMIEPXAYELLOWRUSDPJFFCRUHFBGHGNCXXAAMMVABRHHMSLXXGJXHZXYRGIQQVBUTDJGCWJPSQENZWOCBE
 ROISMHEBDOEMEFTPITRVOGFESMYAITXPSBVZFZTKYMDWFIOEPGQYHQHIVNKSCXTKFULPKYMGRBJVWWNXVXOOONWCWQMAOIGAUROHLQDHNPCQGTHRJSRVYGDZNDBDKMJEZOXMQNMDJWAZJBYAQZPATFHKDOOBPSKMSBSSJBYFVUMYWTFUMPKKLFDGCXPENPWHXGCKZOFYYPORCDMKSXRQPYXONJLHYZQQZLGGU
 TSBZZGHZEXDPCQGKPGYSYWCVDFWTXTYIIQBYKTYHZVBSBOCQPKEPCWVHQTCTBZGKBESYFGIZTIHFDBBNHIXPOIXRADDSGKLSKSYPVOWSROFGHLHPATYWUQVGTOWDWBZLZUEQKUFKPRIFJZQYEZMMRFPAISFSMIJNSIYONVHHZQURMMHMHHZKSURLMGWVDUCRXUTJDTQUUMSENZZMUYJIZTMXTBMTFYMFFCPFF
 WHQHTVPVWPTTQJWFCLWHUWDKVPGXCQRSVBBNCEMESRUAIVIBJUBZUMPYWDQOMQSFTSANYTOBCSQTJKZQGTQLTWUQXTLDSNQPYGXFSZQCZLJAMBEYIQLAUMALGVLUGMMAGEGAPYTMNNZOUZFAGJJYYLHIDHZOSDCVSHVVXJXQMUDHSTKPCIEXTBOITEPQWVCSYDGDTLKEIKPXBZCJAHIMSGPQOSQVXDAMYKZVW
```