In [1]:
```
!pip install boto3
```

```
Requirement already satisfied: boto3 in c:\users\vaish\anaconda3\lib\site-packages
(1.37.17)
Requirement already satisfied: botocore<1.38.0,>=1.37.17 in c:\users\vaish\anaconda3
\lib\site-packages (from boto3) (1.37.17)
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\vaish\anaconda3\li
b\site-packages (from boto3) (1.0.1)
Requirement already satisfied: s3transfer<0.12.0,>=0.11.0 in c:\users\vaish\anaconda
3\lib\site-packages (from boto3) (0.11.4)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\vaish\anacond
a3\lib\site-packages (from botocore<1.38.0,>=1.37.17->boto3) (2.9.0.post0)
Requirement already satisfied: urllib3!=2.2.0,<3,>=1.25.4 in c:\users\vaish\anaconda
3\lib\site-packages (from botocore<1.38.0,>=1.37.17->boto3) (2.2.2)
Requirement already satisfied: six>=1.5 in c:\users\vaish\anaconda3\lib\site-package
s (from python-dateutil<3.0.0,>=2.1->botocore<1.38.0,>=1.37.17->boto3) (1.16.0)
```

In [8]:
```
!aws --version
```

```
'aws' is not recognized as an internal or external command,
operable program or batch file.
```

In [2]:
```
import boto3
s3= boto3.client("s3")
```

In [3]:
```
import boto3

# Create an S3 client

s3 = boto3.client("s3")
```

In [10]:
```
bucket_name = "mynlpmru5"

s3.create_bucket(

    Bucket=bucket_name,

    CreateBucketConfiguration={"LocationConstraint": "eu-north-1"}

)
print(f"Bucket '{bucket_name}' created successfully!")
```

```
---------------------------------------------------------------------------
NoCredentialsError                        Traceback (most recent call last)
Cell In[10], line 3
      1 bucket_name = "mynlpmru5"
----> 3 s3.create_bucket(
      4
      5     Bucket=bucket_name,
      6
      7     CreateBucketConfiguration={"LocationConstraint": "eu-north-1"}
      8
      9 )
     10 print(f"Bucket '{bucket_name}' created successfully!")

File ~\anaconda3\Lib\site-packages\botocore\client.py:570, in ClientCreator._create_
api_method.<locals>._api_call(self, *args, **kwargs)
    566     raise TypeError(
    567         f"{py_operation_name}() only accepts keyword arguments."
    568     )
    569 # The "self" in this scope is referring to the BaseClient.
--> 570 return self._make_api_call(operation_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\client.py:1013, in BaseClient._make_api_
call(self, operation_name, api_params)
   1009     maybe_compress_request(
   1010         self.meta.config, request_dict, operation_model
   1011     )
   1012     apply_request_checksum(request_dict)
-> 1013     http, parsed_response = self._make_request(
   1014         operation_model, request_dict, request_context
   1015     )
   1017 self.meta.events.emit(
   1018     f'after-call.{service_id}.{operation_name}',
   1019     http_response=http,
   (...)
   1022     context=request_context,
   1023 )
   1025 if http.status_code >= 300:

File ~\anaconda3\Lib\site-packages\botocore\client.py:1037, in BaseClient._make_requ
est(self, operation_model, request_dict, request_context)
   1035 def _make_request(self, operation_model, request_dict, request_context):
   1036     try:
-> 1037         return self._endpoint.make_request(operation_model, request_dict)
   1038     except Exception as e:
   1039         self.meta.events.emit(
   1040             f'after-call-error.{self._service_model.service_id.hyphenize()}.
{operation_model.name}',
   1041             exception=e,
   1042             context=request_context,
   1043         )
```

```
File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:119, in Endpoint.make_reques
t(self, operation_model, request_dict)
    113 def make_request(self, operation_model, request_dict):
    114     logger.debug(
    115         "Making request for %s with params: %s",
    116         operation_model,
    117         request_dict,
    118     )
--> 119     return self._send_request(request_dict, operation_model)

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:196, in Endpoint._send_reque
st(self, request_dict, operation_model)
    194 context = request_dict['context']
    195 self._update_retries_context(context, attempts)
--> 196 request = self.create_request(request_dict, operation_model)
    197 success_response, exception = self._get_response(
    198     request, operation_model, context
    199 )
    200 while self._needs_retry(
    201     attempts,
    202     operation_model,
   (...)
    205     exception,
    206 ):

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:132, in Endpoint.create_requ
est(self, params, operation_model)
    130     service_id = operation_model.service_model.service_id.hyphenize()
    131     event_name = f'request-created.{service_id}.{operation_model.name}'
--> 132     self._event_emitter.emit(
    133         event_name,
    134         request=request,
    135         operation_name=operation_model.name,
    136     )
    137 prepared_request = self.prepare_request(request)
    138 return prepared_request

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:412, in EventAliaser.emit(self,
event_name, **kwargs)
    410 def emit(self, event_name, **kwargs):
    411     aliased_event_name = self._alias_event_name(event_name)
--> 412     return self._emitter.emit(aliased_event_name, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:256, in HierarchicalEmitter.emi
t(self, event_name, **kwargs)
    245 def emit(self, event_name, **kwargs):
    246     """
    247     Emit an event by name with arguments passed as keyword args.
    248
   (...)
    254             handlers.
    255     """
--> 256     return self._emit(event_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:239, in HierarchicalEmitter._em
```

```
it(self, event_name, kwargs, stop_on_response)
    237 for handler in handlers_to_call:
    238     logger.debug('Event %s: calling handler %s', event_name, handler)
--> 239     response = handler(**kwargs)
    240     responses.append((handler, response))
    241     if stop_on_response and response is not None:

File ~\anaconda3\Lib\site-packages\botocore\signers.py:106, in RequestSigner.handler
(self, operation_name, request, **kwargs)
    101 def handler(self, operation_name=None, request=None, **kwargs):
    102     # This is typically hooked up to the "request-created" event
    103     # from a client's event emitter.  When a new request is created
    104     # this method is invoked to sign the request.
    105     # Don't call this method directly.
--> 106     return self.sign(operation_name, request)

File ~\anaconda3\Lib\site-packages\botocore\signers.py:198, in RequestSigner.sign(se
lf, operation_name, request, region_name, signing_type, expires_in, signing_name)
    195     else:
    196         raise e
--> 198 auth.add_auth(request)

File ~\anaconda3\Lib\site-packages\botocore\auth.py:424, in SigV4Auth.add_auth(self,
request)
    422 def add_auth(self, request):
    423     if self.credentials is None:
--> 424         raise NoCredentialsError()
    425     datetime_now = datetime.datetime.utcnow()
    426     request.context['timestamp'] = datetime_now.strftime(SIGV4_TIMESTAMP)

NoCredentialsError: Unable to locate credentials
```

In [12]:

```
response = s3.list_buckets()
print("Existing Buckets:")

for bucket in response["Buckets"]:

    print(f"- {bucket['Name']}")
```

```
---------------------------------------------------------------------------
NoCredentialsError                        Traceback (most recent call last)
Cell In[12], line 1
----> 1 response = s3.list_buckets()
      2 print("Existing Buckets:")
      4 for bucket in response["Buckets"]:

File ~\anaconda3\Lib\site-packages\botocore\client.py:570, in ClientCreator._create_
api_method.<locals>._api_call(self, *args, **kwargs)
    566     raise TypeError(
    567         f"{py_operation_name}() only accepts keyword arguments."
    568     )
    569 # The "self" in this scope is referring to the BaseClient.
--> 570 return self._make_api_call(operation_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\client.py:1013, in BaseClient._make_api_
call(self, operation_name, api_params)
   1009     maybe_compress_request(
   1010         self.meta.config, request_dict, operation_model
   1011     )
   1012     apply_request_checksum(request_dict)
-> 1013     http, parsed_response = self._make_request(
   1014         operation_model, request_dict, request_context
   1015     )
   1017 self.meta.events.emit(
   1018     f'after-call.{service_id}.{operation_name}',
   1019     http_response=http,
   (...)
   1022     context=request_context,
   1023 )
   1025 if http.status_code >= 300:

File ~\anaconda3\Lib\site-packages\botocore\client.py:1037, in BaseClient._make_requ
est(self, operation_model, request_dict, request_context)
   1035 def _make_request(self, operation_model, request_dict, request_context):
   1036     try:
-> 1037         return self._endpoint.make_request(operation_model, request_dict)
   1038     except Exception as e:
   1039         self.meta.events.emit(
   1040             f'after-call-error.{self._service_model.service_id.hyphenize()}.
{operation_model.name}',
   1041             exception=e,
   1042             context=request_context,
   1043         )

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:119, in Endpoint.make_reques
t(self, operation_model, request_dict)
    113 def make_request(self, operation_model, request_dict):
    114     logger.debug(
    115         "Making request for %s with params: %s",
```

```
    116         operation_model,
    117         request_dict,
    118     )
--> 119     return self._send_request(request_dict, operation_model)

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:196, in Endpoint._send_reque
st(self, request_dict, operation_model)
    194 context = request_dict['context']
    195 self._update_retries_context(context, attempts)
--> 196 request = self.create_request(request_dict, operation_model)
    197 success_response, exception = self._get_response(
    198     request, operation_model, context
    199 )
    200 while self._needs_retry(
    201     attempts,
    202     operation_model,
   (...)
    205     exception,
    206 ):

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:132, in Endpoint.create_requ
est(self, params, operation_model)
    130     service_id = operation_model.service_model.service_id.hyphenize()
    131     event_name = f'request-created.{service_id}.{operation_model.name}'
--> 132     self._event_emitter.emit(
    133         event_name,
    134         request=request,
    135         operation_name=operation_model.name,
    136     )
    137 prepared_request = self.prepare_request(request)
    138 return prepared_request

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:412, in EventAliaser.emit(self,
event_name, **kwargs)
    410 def emit(self, event_name, **kwargs):
    411     aliased_event_name = self._alias_event_name(event_name)
--> 412     return self._emitter.emit(aliased_event_name, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:256, in HierarchicalEmitter.emi
t(self, event_name, **kwargs)
    245 def emit(self, event_name, **kwargs):
    246     """
    247     Emit an event by name with arguments passed as keyword args.
    248
   (...)
    254             handlers.
    255     """
--> 256     return self._emit(event_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:239, in HierarchicalEmitter._em
it(self, event_name, kwargs, stop_on_response)
    237 for handler in handlers_to_call:
    238     logger.debug('Event %s: calling handler %s', event_name, handler)
--> 239     response = handler(**kwargs)
    240     responses.append((handler, response))
    241     if stop_on_response and response is not None:
```

```
File ~\anaconda3\Lib\site-packages\botocore\signers.py:106, in RequestSigner.handler
(self, operation_name, request, **kwargs)
    101 def handler(self, operation_name=None, request=None, **kwargs):
    102     # This is typically hooked up to the "request-created" event
    103     # from a client's event emitter.  When a new request is created
    104     # this method is invoked to sign the request.
    105     # Don't call this method directly.
--> 106     return self.sign(operation_name, request)

File ~\anaconda3\Lib\site-packages\botocore\signers.py:198, in RequestSigner.sign(se
lf, operation_name, request, region_name, signing_type, expires_in, signing_name)
    195     else:
    196         raise e
--> 198 auth.add_auth(request)

File ~\anaconda3\Lib\site-packages\botocore\auth.py:424, in SigV4Auth.add_auth(self,
request)
    422 def add_auth(self, request):
    423     if self.credentials is None:
--> 424         raise NoCredentialsError()
    425     datetime_now = datetime.datetime.utcnow()
    426     request.context['timestamp'] = datetime_now.strftime(SIGV4_TIMESTAMP)

NoCredentialsError: Unable to locate credentials
```

In [17]:

```
# Upload a file

s3.upload_file("test.txt", "mynlpmru", "data/testfile1.txt")
print("Upload complete!")
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[17], line 3
      1 # Upload a file
----> 3 s3.upload_file("test.txt", "mynlpmru", "data/testfile1.txt")
      4 print("Upload complete!")

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\boto3\s3\inject.py:170, in upload_file(self, File
name, Bucket, Key, ExtraArgs, Callback, Config)
    135 """Upload a file to an S3 object.
    136
    137 Usage::
   (...)
    167     transfer.
    168 """
    169 with S3Transfer(self, Config) as transfer:
--> 170     return transfer.upload_file(
    171         filename=Filename,
    172         bucket=Bucket,
    173         key=Key,
    174         extra_args=ExtraArgs,
    175         callback=Callback,
    176     )

File ~\anaconda3\Lib\site-packages\boto3\s3\transfer.py:372, in S3Transfer.upload_fi
le(self, filename, bucket, key, callback, extra_args)
    368 future = self._manager.upload(
    369     filename, bucket, key, extra_args, subscribers
    370 )
    371 try:
--> 372     future.result()
    373 # If a client error was raised, add the backwards compatibility layer
    374 # that raises a S3UploadFailedError. These specific errors were only
    375 # ever thrown for upload_parts but now can be thrown for any related
    376 # client error.
    377 except ClientError as e:

File ~\anaconda3\Lib\site-packages\s3transfer\futures.py:111, in TransferFuture.resu
lt(self)
    106 def result(self):
    107     try:
    108         # Usually the result() method blocks until the transfer is done,
    109         # however if a KeyboardInterrupt is raised we want want to exit
    110         # out of this and propagate the exception.
--> 111         return self._coordinator.result()
    112     except KeyboardInterrupt as e:
    113         self.cancel()

File ~\anaconda3\Lib\site-packages\s3transfer\futures.py:272, in TransferCoordinato
r.result(self)
```

```
    269 # Once done waiting, raise an exception if present or return the
    270 # final result.
    271 if self._exception:
--> 272     raise self._exception
    273 return self._result

File ~\anaconda3\Lib\site-packages\s3transfer\tasks.py:272, in SubmissionTask._main
(self, transfer_future, **kwargs)
    268     self._transfer_coordinator.set_status_to_running()
    270     # Call the submit method to start submitting tasks to execute the
    271     # transfer.
--> 272     self._submit(transfer_future=transfer_future, **kwargs)
    273 except BaseException as e:
    274     # If there was an exception raised during the submission of task
    275     # there is a chance that the final task that signals if a transfer
   (...)
    284
    285     # Set the exception, that caused the process to fail.
    286     self._log_and_set_exception(e)

File ~\anaconda3\Lib\site-packages\s3transfer\upload.py:596, in UploadSubmissionTas
k._submit(self, client, config, osutil, request_executor, transfer_future, bandwidth
_limiter)
    594 # Determine the size if it was not provided
    595 if transfer_future.meta.size is None:
--> 596     upload_input_manager.provide_transfer_size(transfer_future)
    598 # Do a multipart upload if needed, otherwise do a regular put object.
    599 if not upload_input_manager.requires_multipart_upload(
    600     transfer_future, config
    601 ):

File ~\anaconda3\Lib\site-packages\s3transfer\upload.py:245, in UploadFilenameInputM
anager.provide_transfer_size(self, transfer_future)
    243 def provide_transfer_size(self, transfer_future):
    244     transfer_future.meta.provide_transfer_size(
--> 245         self._osutil.get_file_size(transfer_future.meta.call_args.fileobj)
    246     )

File ~\anaconda3\Lib\site-packages\s3transfer\utils.py:262, in OSUtils.get_file_size
(self, filename)
    261 def get_file_size(self, filename):
--> 262     return os.path.getsize(filename)

File <frozen genericpath>:62, in getsize(filename)

FileNotFoundError: [WinError 2] The system cannot find the file specified: 'test.tx
t'
```

In [19]:

```
s3.download_file("mynlpmru", "data/testfile1.txt", "downloaded_test2.txt")

print("Download successful!")
```

```
---------------------------------------------------------------------------
NoCredentialsError                        Traceback (most recent call last)
Cell In[19], line 1
----> 1 s3.download_file("mynlpmru", "data/testfile1.txt", "downloaded_test2.txt")
      3 print("Download successful!")

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\boto3\s3\inject.py:218, in download_file(self, Bu
cket, Key, Filename, ExtraArgs, Callback, Config)
    183 """Download an S3 object to a file.
    184
    185 Usage::
   (...)
    215     transfer.
    216 """
    217 with S3Transfer(self, Config) as transfer:
--> 218     return transfer.download_file(
    219         bucket=Bucket,
    220         key=Key,
    221         filename=Filename,
    222         extra_args=ExtraArgs,
    223         callback=Callback,
    224     )

File ~\anaconda3\Lib\site-packages\boto3\s3\transfer.py:406, in S3Transfer.download_
file(self, bucket, key, filename, extra_args, callback)
    402 future = self._manager.download(
    403     bucket, key, filename, extra_args, subscribers
    404 )
    405 try:
--> 406     future.result()
    407 # This is for backwards compatibility where when retries are
    408 # exceeded we need to throw the same error from boto3 instead of
    409 # s3transfer's built in RetriesExceededError as current users are
    410 # catching the boto3 one instead of the s3transfer exception to do
    411 # their own retries.
    412 except S3TransferRetriesExceededError as e:

File ~\anaconda3\Lib\site-packages\s3transfer\futures.py:111, in TransferFuture.resu
lt(self)
    106 def result(self):
    107     try:
    108         # Usually the result() method blocks until the transfer is done,
    109         # however if a KeyboardInterrupt is raised we want want to exit
    110         # out of this and propagate the exception.
--> 111         return self._coordinator.result()
    112     except KeyboardInterrupt as e:
    113         self.cancel()

File ~\anaconda3\Lib\site-packages\s3transfer\futures.py:272, in TransferCoordinato
r.result(self)
```

```
    269 # Once done waiting, raise an exception if present or return the
    270 # final result.
    271 if self._exception:
--> 272     raise self._exception
    273 return self._result

File ~\anaconda3\Lib\site-packages\s3transfer\tasks.py:272, in SubmissionTask._main
(self, transfer_future, **kwargs)
    268     self._transfer_coordinator.set_status_to_running()
    270     # Call the submit method to start submitting tasks to execute the
    271     # transfer.
--> 272     self._submit(transfer_future=transfer_future, **kwargs)
    273 except BaseException as e:
    274     # If there was an exception raised during the submission of task
    275     # there is a chance that the final task that signals if a transfer
   (...)
    284
    285     # Set the exception, that caused the process to fail.
    286     self._log_and_set_exception(e)

File ~\anaconda3\Lib\site-packages\s3transfer\download.py:352, in DownloadSubmission
Task._submit(self, client, config, osutil, request_executor, io_executor, transfer_f
uture, bandwidth_limiter)
    323 """
    324 :param client: The client associated with the transfer manager
    325
   (...)
    347     downloading streams
    348 """
    349 if transfer_future.meta.size is None:
    350     # If a size was not provided figure out the size for the
    351     # user.
--> 352     response = client.head_object(
    353         Bucket=transfer_future.meta.call_args.bucket,
    354         Key=transfer_future.meta.call_args.key,
    355         **transfer_future.meta.call_args.extra_args,
    356     )
    357     transfer_future.meta.provide_transfer_size(
    358         response['ContentLength']
    359     )
    361 download_output_manager = self._get_download_output_manager_cls(
    362     transfer_future, osutil
    363 )(osutil, self._transfer_coordinator, io_executor)

File ~\anaconda3\Lib\site-packages\botocore\client.py:570, in ClientCreator._create_
api_method.<locals>._api_call(self, *args, **kwargs)
    566     raise TypeError(
    567         f"{py_operation_name}() only accepts keyword arguments."
    568     )
    569 # The "self" in this scope is referring to the BaseClient.
--> 570 return self._make_api_call(operation_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
```

```
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\client.py:1013, in BaseClient._make_api_
call(self, operation_name, api_params)
   1009     maybe_compress_request(
   1010         self.meta.config, request_dict, operation_model
   1011     )
   1012     apply_request_checksum(request_dict)
-> 1013     http, parsed_response = self._make_request(
   1014         operation_model, request_dict, request_context
   1015     )
   1017 self.meta.events.emit(
   1018     f'after-call.{service_id}.{operation_name}',
   1019     http_response=http,
   (...)
   1022     context=request_context,
   1023 )
   1025 if http.status_code >= 300:

File ~\anaconda3\Lib\site-packages\botocore\client.py:1037, in BaseClient._make_requ
est(self, operation_model, request_dict, request_context)
   1035 def _make_request(self, operation_model, request_dict, request_context):
   1036     try:
-> 1037         return self._endpoint.make_request(operation_model, request_dict)
   1038     except Exception as e:
   1039         self.meta.events.emit(
   1040             f'after-call-error.{self._service_model.service_id.hyphenize()}.
{operation_model.name}',
   1041             exception=e,
   1042             context=request_context,
   1043         )

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:119, in Endpoint.make_reques
t(self, operation_model, request_dict)
    113 def make_request(self, operation_model, request_dict):
    114     logger.debug(
    115         "Making request for %s with params: %s",
    116         operation_model,
    117         request_dict,
    118     )
--> 119     return self._send_request(request_dict, operation_model)

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:196, in Endpoint._send_reque
st(self, request_dict, operation_model)
    194 context = request_dict['context']
    195 self._update_retries_context(context, attempts)
--> 196 request = self.create_request(request_dict, operation_model)
    197 success_response, exception = self._get_response(
    198     request, operation_model, context
    199 )
    200 while self._needs_retry(
    201     attempts,
    202     operation_model,
   (...)
    205     exception,
    206 ):
```

```
File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:132, in Endpoint.create_request(self, params, operation_model)
    130     service_id = operation_model.service_model.service_id.hyphenize()
    131     event_name = f'request-created.{service_id}.{operation_model.name}'
--> 132     self._event_emitter.emit(
    133         event_name,
    134         request=request,
    135         operation_name=operation_model.name,
    136     )
    137 prepared_request = self.prepare_request(request)
    138 return prepared_request

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:412, in EventAliaser.emit(self, event_name, **kwargs)
    410 def emit(self, event_name, **kwargs):
    411     aliased_event_name = self._alias_event_name(event_name)
--> 412     return self._emitter.emit(aliased_event_name, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:256, in HierarchicalEmitter.emit(self, event_name, **kwargs)
    245 def emit(self, event_name, **kwargs):
    246     """
    247     Emit an event by name with arguments passed as keyword args.
    248
   (...)
    254             handlers.
    255     """
--> 256     return self._emit(event_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:239, in HierarchicalEmitter._emit(self, event_name, kwargs, stop_on_response)
    237 for handler in handlers_to_call:
    238     logger.debug('Event %s: calling handler %s', event_name, handler)
--> 239     response = handler(**kwargs)
    240     responses.append((handler, response))
    241     if stop_on_response and response is not None:

File ~\anaconda3\Lib\site-packages\botocore\signers.py:106, in RequestSigner.handler(self, operation_name, request, **kwargs)
    101 def handler(self, operation_name=None, request=None, **kwargs):
    102     # This is typically hooked up to the "request-created" event
    103     # from a client's event emitter.  When a new request is created
    104     # this method is invoked to sign the request.
    105     # Don't call this method directly.
--> 106     return self.sign(operation_name, request)

File ~\anaconda3\Lib\site-packages\botocore\signers.py:198, in RequestSigner.sign(self, operation_name, request, region_name, signing_type, expires_in, signing_name)
    195     else:
    196         raise e
--> 198 auth.add_auth(request)

File ~\anaconda3\Lib\site-packages\botocore\auth.py:424, in SigV4Auth.add_auth(self, request)
    422 def add_auth(self, request):
```

```
    423     if self.credentials is None:
--> 424         raise NoCredentialsError()
    425     datetime_now = datetime.datetime.utcnow()
    426     request.context['timestamp'] = datetime_now.strftime(SIGV4_TIMESTAMP)

NoCredentialsError: Unable to locate credentials
```

In [21]:

```
response = s3.list_objects_v2(Bucket="mynlpmru")



print("Files in bucket:")

if "Contents" in response:

    for obj in response["Contents"]:

        print(f"- {obj['Key']} (Size: {obj['Size']} bytes)")

else:

    print("Bucket is empty.")
```

```
---------------------------------------------------------------------------
NoCredentialsError                        Traceback (most recent call last)
Cell In[21], line 1
----> 1 response = s3.list_objects_v2(Bucket="mynlpmru")
      5 print("Files in bucket:")
      7 if "Contents" in response:

File ~\anaconda3\Lib\site-packages\botocore\client.py:570, in ClientCreator._create_
api_method.<locals>._api_call(self, *args, **kwargs)
    566     raise TypeError(
    567         f"{py_operation_name}() only accepts keyword arguments."
    568     )
    569 # The "self" in this scope is referring to the BaseClient.
--> 570 return self._make_api_call(operation_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\client.py:1013, in BaseClient._make_api_
call(self, operation_name, api_params)
   1009     maybe_compress_request(
   1010         self.meta.config, request_dict, operation_model
   1011     )
   1012     apply_request_checksum(request_dict)
-> 1013     http, parsed_response = self._make_request(
   1014         operation_model, request_dict, request_context
   1015     )
   1017 self.meta.events.emit(
   1018     f'after-call.{service_id}.{operation_name}',
   1019     http_response=http,
   (...)
   1022     context=request_context,
   1023 )
   1025 if http.status_code >= 300:

File ~\anaconda3\Lib\site-packages\botocore\client.py:1037, in BaseClient._make_requ
est(self, operation_model, request_dict, request_context)
   1035 def _make_request(self, operation_model, request_dict, request_context):
   1036     try:
-> 1037         return self._endpoint.make_request(operation_model, request_dict)
   1038     except Exception as e:
   1039         self.meta.events.emit(
   1040             f'after-call-error.{self._service_model.service_id.hyphenize()}.
{operation_model.name}',
   1041             exception=e,
   1042             context=request_context,
   1043         )

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:119, in Endpoint.make_reques
t(self, operation_model, request_dict)
    113 def make_request(self, operation_model, request_dict):
    114     logger.debug(
    115         "Making request for %s with params: %s",
```

```
    116         operation_model,
    117         request_dict,
    118     )
--> 119     return self._send_request(request_dict, operation_model)

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:196, in Endpoint._send_reque
st(self, request_dict, operation_model)
    194 context = request_dict['context']
    195 self._update_retries_context(context, attempts)
--> 196 request = self.create_request(request_dict, operation_model)
    197 success_response, exception = self._get_response(
    198     request, operation_model, context
    199 )
    200 while self._needs_retry(
    201     attempts,
    202     operation_model,
   (...)
    205     exception,
    206 ):

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:132, in Endpoint.create_requ
est(self, params, operation_model)
    130     service_id = operation_model.service_model.service_id.hyphenize()
    131     event_name = f'request-created.{service_id}.{operation_model.name}'
--> 132     self._event_emitter.emit(
    133         event_name,
    134         request=request,
    135         operation_name=operation_model.name,
    136     )
    137 prepared_request = self.prepare_request(request)
    138 return prepared_request

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:412, in EventAliaser.emit(self,
event_name, **kwargs)
    410 def emit(self, event_name, **kwargs):
    411     aliased_event_name = self._alias_event_name(event_name)
--> 412     return self._emitter.emit(aliased_event_name, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:256, in HierarchicalEmitter.emi
t(self, event_name, **kwargs)
    245 def emit(self, event_name, **kwargs):
    246     """
    247     Emit an event by name with arguments passed as keyword args.
    248
   (...)
    254             handlers.
    255     """
--> 256     return self._emit(event_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:239, in HierarchicalEmitter._em
it(self, event_name, kwargs, stop_on_response)
    237 for handler in handlers_to_call:
    238     logger.debug('Event %s: calling handler %s', event_name, handler)
--> 239     response = handler(**kwargs)
    240     responses.append((handler, response))
    241     if stop_on_response and response is not None:
```

```
File ~\anaconda3\Lib\site-packages\botocore\signers.py:106, in RequestSigner.handler
(self, operation_name, request, **kwargs)
    101 def handler(self, operation_name=None, request=None, **kwargs):
    102     # This is typically hooked up to the "request-created" event
    103     # from a client's event emitter.  When a new request is created
    104     # this method is invoked to sign the request.
    105     # Don't call this method directly.
--> 106     return self.sign(operation_name, request)

File ~\anaconda3\Lib\site-packages\botocore\signers.py:198, in RequestSigner.sign(se
lf, operation_name, request, region_name, signing_type, expires_in, signing_name)
    195     else:
    196         raise e
--> 198 auth.add_auth(request)

File ~\anaconda3\Lib\site-packages\botocore\auth.py:424, in SigV4Auth.add_auth(self,
request)
    422 def add_auth(self, request):
    423     if self.credentials is None:
--> 424         raise NoCredentialsError()
    425     datetime_now = datetime.datetime.utcnow()
    426     request.context['timestamp'] = datetime_now.strftime(SIGV4_TIMESTAMP)

NoCredentialsError: Unable to locate credentials
```

In [23]:

```
s3.delete_object(Bucket="mynlpmru", Key="data/testfile1.txt")

print("File deleted successfully!")
```

```
---------------------------------------------------------------------------
NoCredentialsError                        Traceback (most recent call last)
Cell In[23], line 1
----> 1 s3.delete_object(Bucket="mynlpmru", Key="data/testfile1.txt")
      3 print("File deleted successfully!")

File ~\anaconda3\Lib\site-packages\botocore\client.py:570, in ClientCreator._create_
api_method.<locals>._api_call(self, *args, **kwargs)
    566     raise TypeError(
    567         f"{py_operation_name}() only accepts keyword arguments."
    568     )
    569 # The "self" in this scope is referring to the BaseClient.
--> 570 return self._make_api_call(operation_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\context.py:124, in with_current_context.
<locals>.decorator.<locals>.wrapper(*args, **kwargs)
    122 if hook:
    123     hook()
--> 124 return func(*args, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\client.py:1013, in BaseClient._make_api_
call(self, operation_name, api_params)
   1009     maybe_compress_request(
   1010         self.meta.config, request_dict, operation_model
   1011     )
   1012     apply_request_checksum(request_dict)
-> 1013     http, parsed_response = self._make_request(
   1014         operation_model, request_dict, request_context
   1015     )
   1017 self.meta.events.emit(
   1018     f'after-call.{service_id}.{operation_name}',
   1019     http_response=http,
   (...)
   1022     context=request_context,
   1023 )
   1025 if http.status_code >= 300:

File ~\anaconda3\Lib\site-packages\botocore\client.py:1037, in BaseClient._make_requ
est(self, operation_model, request_dict, request_context)
   1035 def _make_request(self, operation_model, request_dict, request_context):
   1036     try:
-> 1037         return self._endpoint.make_request(operation_model, request_dict)
   1038     except Exception as e:
   1039         self.meta.events.emit(
   1040             f'after-call-error.{self._service_model.service_id.hyphenize()}.
{operation_model.name}',
   1041             exception=e,
   1042             context=request_context,
   1043         )

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:119, in Endpoint.make_reques
t(self, operation_model, request_dict)
    113 def make_request(self, operation_model, request_dict):
    114     logger.debug(
    115         "Making request for %s with params: %s",
    116         operation_model,
```

```
    117         request_dict,
    118     )
--> 119     return self._send_request(request_dict, operation_model)

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:196, in Endpoint._send_reque
st(self, request_dict, operation_model)
    194 context = request_dict['context']
    195 self._update_retries_context(context, attempts)
--> 196 request = self.create_request(request_dict, operation_model)
    197 success_response, exception = self._get_response(
    198     request, operation_model, context
    199 )
    200 while self._needs_retry(
    201     attempts,
    202     operation_model,
   (...)
    205     exception,
    206 ):

File ~\anaconda3\Lib\site-packages\botocore\endpoint.py:132, in Endpoint.create_requ
est(self, params, operation_model)
    130     service_id = operation_model.service_model.service_id.hyphenize()
    131     event_name = f'request-created.{service_id}.{operation_model.name}'
--> 132     self._event_emitter.emit(
    133         event_name,
    134         request=request,
    135         operation_name=operation_model.name,
    136     )
    137 prepared_request = self.prepare_request(request)
    138 return prepared_request

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:412, in EventAliaser.emit(self,
event_name, **kwargs)
    410 def emit(self, event_name, **kwargs):
    411     aliased_event_name = self._alias_event_name(event_name)
--> 412     return self._emitter.emit(aliased_event_name, **kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:256, in HierarchicalEmitter.emi
t(self, event_name, **kwargs)
    245 def emit(self, event_name, **kwargs):
    246     """
    247     Emit an event by name with arguments passed as keyword args.
    248
   (...)
    254             handlers.
    255     """
--> 256     return self._emit(event_name, kwargs)

File ~\anaconda3\Lib\site-packages\botocore\hooks.py:239, in HierarchicalEmitter._em
it(self, event_name, kwargs, stop_on_response)
    237 for handler in handlers_to_call:
    238     logger.debug('Event %s: calling handler %s', event_name, handler)
--> 239     response = handler(**kwargs)
    240     responses.append((handler, response))
    241     if stop_on_response and response is not None:
```

```
File ~\anaconda3\Lib\site-packages\botocore\signers.py:106, in RequestSigner.handler
(self, operation_name, request, **kwargs)
    101 def handler(self, operation_name=None, request=None, **kwargs):
    102     # This is typically hooked up to the "request-created" event
    103     # from a client's event emitter.  When a new request is created
    104     # this method is invoked to sign the request.
    105     # Don't call this method directly.
--> 106     return self.sign(operation_name, request)

File ~\anaconda3\Lib\site-packages\botocore\signers.py:198, in RequestSigner.sign(se
lf, operation_name, request, region_name, signing_type, expires_in, signing_name)
    195     else:
    196         raise e
--> 198 auth.add_auth(request)

File ~\anaconda3\Lib\site-packages\botocore\auth.py:424, in SigV4Auth.add_auth(self,
request)
    422 def add_auth(self, request):
    423     if self.credentials is None:
--> 424         raise NoCredentialsError()
    425     datetime_now = datetime.datetime.utcnow()
    426     request.context['timestamp'] = datetime_now.strftime(SIGV4_TIMESTAMP)

NoCredentialsError: Unable to locate credentials
```

In [ ]:
```
s3.put_bucket_versioning(

    Bucket="mynlpmru",

    VersioningConfiguration={"Status": "Enabled"}

)

print("Bucket versioning enabled.")
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: