

try:

https://github.com/podaac/tutorials/blob/master/notebooks/meetings_workshops/workshop_osm_20

```
In [2]: import os
import glob
import s3fs
import requests
import numpy as np
import pandas as pd
import xarray as xr
import hvplot.xarray
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import cartopy
from datetime import datetime
from os.path import isfile, basename, abspath
```

```
In [3]: s3_cred_endpoint = 'https://archive.podaac.earthdata.nasa.gov/s3credentials'
```

```
In [4]: def get_temp_creds():
temp_creds_url = s3_cred_endpoint
return requests.get(temp_creds_url).json()
```

```
In [5]: temp_creds_req = get_temp_creds()
```

```
In [10]: #temp_creds_req
```

```
In [6]: fs_s3 = s3fs.S3FileSystem(anon=False,
key=temp_creds_req['accessKeyId'],
secret=temp_creds_req['secretAccessKey'],
token=temp_creds_req['sessionToken'],
client_kwargs={'region_name': 'us-west-2'})
```

```
In [7]: url='s3://podaac-ops-cumulus-protected/TELLUS_GRAC-GRFO_MASCON_CRI_GRID_RL06_V2/'
```

```
In [12]: s3_file_obj = fs_s3.open(url, mode='rb')
```

```
-----
ClientError                                Traceback (most recent call last)
~/mambaforge/lib/python3.9/site-packages/s3fs/core.py in _call_s3(self, method,
*akwarglist, **kwargs)
    260         try:
```

```

--> 261             out = await method(**additional_kwargs)
    262             return out

~/mambaforge/lib/python3.9/site-packages/aiobotocore/client.py in _make_api_call
(self, operation_name, api_params)
    186             error_class = self.exceptions.from_code(error_code)
--> 187             raise error_class(parsed_response, operation_name)
    188         else:

```

ClientError: An error occurred (403) when calling the HeadObject operation: Forbidden

The above exception was the direct cause of the following exception:

```

PermissionError                               Traceback (most recent call last)
/var/folders/4j/5w3_gh9d0fg3p5l7ltljg11h0000gq/T/ipykernel_20125/1716990729.py i
n <module>
----> 1 s3_file_obj = fs_s3.open(url, mode='rb')

```

```

~/mambaforge/lib/python3.9/site-packages/fsspec/spec.py in open(self, path, mod
e, block_size, cache_options, compression, **kwargs)
    1007         else:
    1008             ac = kwargs.pop("autocommit", not self._intrans)
-> 1009             f = self._open(
    1010                 path,
    1011                 mode=mode,

```

```

~/mambaforge/lib/python3.9/site-packages/s3fs/core.py in _open(self, path, mode,
block_size, acl, version_id, fill_cache, cache_type, autocommit, requester_pays,
cache_options, **kwargs)
    550             cache_type = self.default_cache_type
    551
--> 552             return S3File(
    553                 self,
    554                 path,

```

```

~/mambaforge/lib/python3.9/site-packages/s3fs/core.py in __init__(self, s3, pat
h, mode, block_size, acl, version_id, fill_cache, s3_additional_kwargs, autocomm
it, cache_type, requester_pays, cache_options)
    1800             self.details = s3.info(path)
    1801             self.version_id = self.details.get("VersionId")
-> 1802             super().__init__(
    1803                 s3,
    1804                 path,

```

```

~/mambaforge/lib/python3.9/site-packages/fsspec/spec.py in __init__(self, fs, pa
th, mode, block_size, autocommit, cache_type, cache_options, size, **kwargs)
    1349             self.size = size
    1350         else:
-> 1351             self.size = self.details["size"]
    1352             self.cache = caches[cache_type](
    1353                 self.blocksize, self._fetch_range, self.size, **cache_op
tions

```

```

~/mambaforge/lib/python3.9/site-packages/fsspec/spec.py in details(self)
    1362     def details(self):
    1363         if self._details is None:
-> 1364             self._details = self.fs.info(self.path)
    1365         return self._details
    1366

```

```
~/mambaforge/lib/python3.9/site-packages/fsspec/asyn.py in wrapper(*args, **kwargs)
    89     def wrapper(*args, **kwargs):
    90         self = obj or args[0]
--> 91         return sync(self.loop, func, *args, **kwargs)
    92
    93     return wrapper

~/mambaforge/lib/python3.9/site-packages/fsspec/asyn.py in sync(loop, func, timeout, *args, **kwargs)
    69         raise FSTimeoutError from return_result
    70     elif isinstance(return_result, BaseException):
--> 71         raise return_result
    72     else:
    73         return return_result

~/mambaforge/lib/python3.9/site-packages/fsspec/asyn.py in _runner(event, coro, result, timeout)
    23         coro = asyncio.wait_for(coro, timeout=timeout)
    24     try:
--> 25         result[0] = await coro
    26     except Exception as ex:
    27         result[0] = ex

~/mambaforge/lib/python3.9/site-packages/s3fs/core.py in _info(self, path, bucket, key, refresh, version_id)
   1027         if key:
   1028             try:
-> 1029                 out = await self._call_s3(
   1030                     "head_object",
   1031                     self.kwargs,

~/mambaforge/lib/python3.9/site-packages/s3fs/core.py in _call_s3(self, method, *akwarglist, **kwargs)
    279         err = e
    280         err = translate_boto_error(err)
--> 281         raise err
    282
    283     call_s3 = sync_wrapper(_call_s3)
```

PermissionError: Forbidden

In []: