introduction

This document describes:

- creating an operating environment suitable for using the S3-subsetting example use-case Jupyter notebooks titled `ps1_cutouts.ipynb` and `ps1_galex_stacking.ipynb`
- obtaining and running the example notebooks in that environment

an important note on scope

This is not a guide for using subsetting techniques for practical purposes. Most of the setup instructions in this document support specific needs of the example notebooks. The example notebooks are intended to provide proof-of-concept operations and technology demonstration. Some *scientific accuracy* has been sacrificed (e.g. incomplete QA of outputs) in service to providing a proof-of-concept, and you should not draw scientific inferences from these outputs without further refinement.

setup

step 1: make an instance

Create an EC2 instance in us-east-1 with the following parameters:

- AWS Ubuntu 22.04 (Jammy) LTS AMI
- t3.small instance type
- 8 GB gp3 EBS root volume

Make sure that the security group you use for this instance allows SSH access from your IP.

notes

- I specify us-east-1 because all of the objects the currently-written benchmarks access are in buckets in us-east-1, so the bucket owners will incur egress costs if you access them from a different AWS Region. There is nothing else special about us-east-1. If you would like to write a different set of benchmarks that access objects in a different AWS Region, you should create an instance there instead.
- **Please note that Million Concepts is unable to make the products it staged for this project available to the public. STScI may make them publicly available at some point in the future. All of the *source* data are publicly available, and this product structure can be recreated. Please contact us if you have questions on this topic.**
- The instructions below specify use of superuser privileges for a variety of purposes. This is mostly not strictly necessary. You can install *goofys* into non-privileged directories, so long as they are in your working path. Other than that, you could install and use this suite in an environment in which you do not have superuser privileges, so long as it has not been specially configured so that users cannot mount and dismount FUSE filesystems and access

device information (like TIKE). However, I strongly recommend just creating an EC2 instance you control.

- I specify Ubuntu 22.04 LTS because it is the highest Ubuntu version with an officially-supported AMI at the time of writing (2022-08-16). I do not anticipate any forward compatibility issues, so if you are using this guide after the release of a new official Ubuntu AMI, use the newer AMI instead.
- The notebooks will most likely work on other Linux distributions, although I have not tested it. Some paths, instructions to package managers, etc. might need to be changed, depending on the distribution. It could also be made to work on MacOS with fairly small modifications.
- I specify t3.small with 8 Gb local EBS storage because it is just about the smallest / cheapest instance that can successfully hold and run the environment and notebooks. Many other instance types will work, and should not change performance very much.

step 2: install support software

1. SSH to the instance and update the stock OS software:
    a. *sudo apt update && sudo apt upgrade*
2. Install and initialize your preferred version of *conda.*
    a. I like Mambaforge (https://github.com/conda-forge/miniforge) but this is not mandatory. I would not recommend Anaconda, because it will install many unnecessary packages that will take up space for no reason.
    b. make sure *conda* is accessible from your path. The installer will run *conda init* by default, but you will need to log out and back in, open a new shell, source your rcfile, or something like that in order for it to be accessible immediately after installation.
3. install goofys (used for FUSE mounts)
    a. *wget https://github.com/kahing/goofys/releases/latest/download/goofys*
    b. make it executable and place/link it somewhere that will be in the “ubuntu” user’s path. for example:
        i. *chmod +x goofys && sudo mv goofys /usr/bin/goofys*
4. clone the fornax-s3-subsets repo:
    a. *git clone https://github.com/fornax-navo/fornax-s3-subsets.git*
5. Create the “fornax-usecase” *conda* environment using the fornax-s3-subsets/subset/usecase_environment.yml file:
    a. *cd fornax-s3-subsets/subset && mamba env create -f bench_environment.yml* (or *conda env…* if you didn’t install Mambaforge)
    b. important: make sure you have this environment active when you execute any code from the benchmark suite (*conda activate fornax-bench*), or else explicitly pass paths to that environment’s interpreter(s).
6. reboot (*sudo reboot)* and SSH back into the instance.

step 3: set up paths and credentials

1. Set up AWS client configuration

a. Add your appropriate AWS credentials to the machine by either running `aws config` from within the active `conda` environment or (*advanced*) creating ~/.aws/config file and ~/.aws/credentials files.
b. Define us-east-1 as a default AWS region in ~/.aws/config so that *fsspec* and *goofys* will be able to find the buckets. I also recommend setting the default AWS output style to JSON. If you ran `aws config`, then this file was already created. Here is a minimal example of what the config file should look like:

```
[default]
region = us-east-1
output = json
```

2. (Optional) Make a mountpoint for *goofys*. The notebooks automatically generate a mount point at ~/s3 by default. If you're ok with the default, you don't need to do anything else. You can change this by modifying the “mountpoint” value in the SETTINGS variable in the notebooks. It must be somewhere that you have write access to.

execution

Now you are fully prepared to use the suite to run S3-subsetting use case notebooks. I recommend tunneling to a Jupyter server on your instance and executing benchmarks from that Notebook. (If you’d like to do this but don’t have an established workflow for it, ping me and I’ll help you out.) If you prefer, however, you can extract code from that Notebook and execute it however you like.