Unlocking Large Datasets on Cloud-Native Data Workflows

Lessons on using Icechunk, Virtualizar, and Kerchunk to enable Single-Point Access to Distributed Cloud Files through xarray

GOBIERNO DE ESPAÑA

VICEPRESIDENCIA
TERCERA DEL GOBIERNO

MINISTERIO
PARA LA TRANSICIÓN ECOLÓGICA
Y EL RETO DEMOGRÁFICO

aemet
Agencia Estatal de Meteorología

EUMETSAT
NWC SAF

GOBIERNO
DE ESPAÑA

VICEPRESIDENCIA
TERCERA DEL GOBIERNO

MINISTERIO
PARA LA TRANSICIÓN ECOLÓGICA
Y EL RETO DEMOGRÁFICO

aemet
Agencia Estatal de Meteorología

EUMETSAT
NWC SAF

Accessing NWC SAF CF buckets with kerchunk, virtualizarr & icechunk.

Short guide for: https://gitlab.aemet.es/jllisov/vzarr.git

jupyter

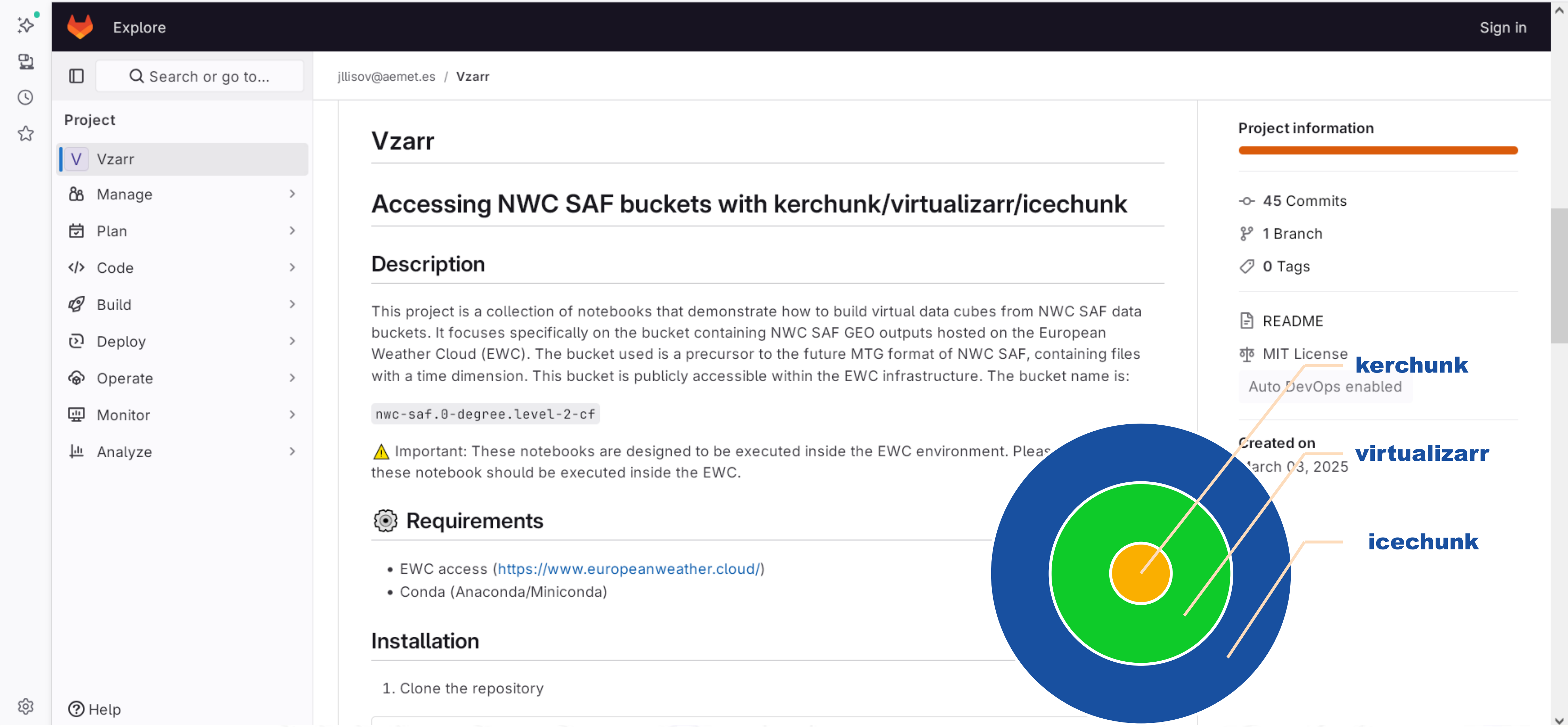
GOBIERNO
DE ESPAÑA

VICEPRESIDENCIA
TERCERA DEL GOBIERNO

MINISTERIO
PARA LA TRANSICIÓN ECOLÓGICA
Y EL RETO DEMOGRÁFICO

aemet
Agencia Estatal de Meteorología

EUMETSAT
NWC SAF



Vzarr

Accessing NWC SAF buckets with kerchunk/virtualizarr/icechunk

Description

This project is a collection of notebooks that demonstrate how to build virtual data cubes from NWC SAF data buckets. It focuses specifically on the bucket containing NWC SAF GEO outputs hosted on the European Weather Cloud (EWC). The bucket used is a precursor to the future MTG format of NWC SAF, containing files with a time dimension. This bucket is publicly accessible within the EWC infrastructure. The bucket name is:

```
nwc-saf.0-degree.level-2-cf
```

⚠ Important: These notebooks are designed to be executed inside the EWC environment. Pleas
these notebook should be executed inside the EWC.

⚙ Requirements

- EWC access (https://www.europeanweather.cloud/)
- Conda (Anaconda/Miniconda)

Installation

1. Clone the repository

The project is a set of notebooks teaching how to build virtual data cubes from NWC SAF CF buckets.

```
Virtual access implies not duplicating the
       files to build the data cube.
```

The format of the files (for the lessons) is the precursor for MTG format of the NWC SAF.
This bucket is public inside the EWC (***"nwc-saf.0-degree.level-2-cf"***)

```
The full lessons can only be executed on EWC.
```

GOBIERNO DE ESPAÑA

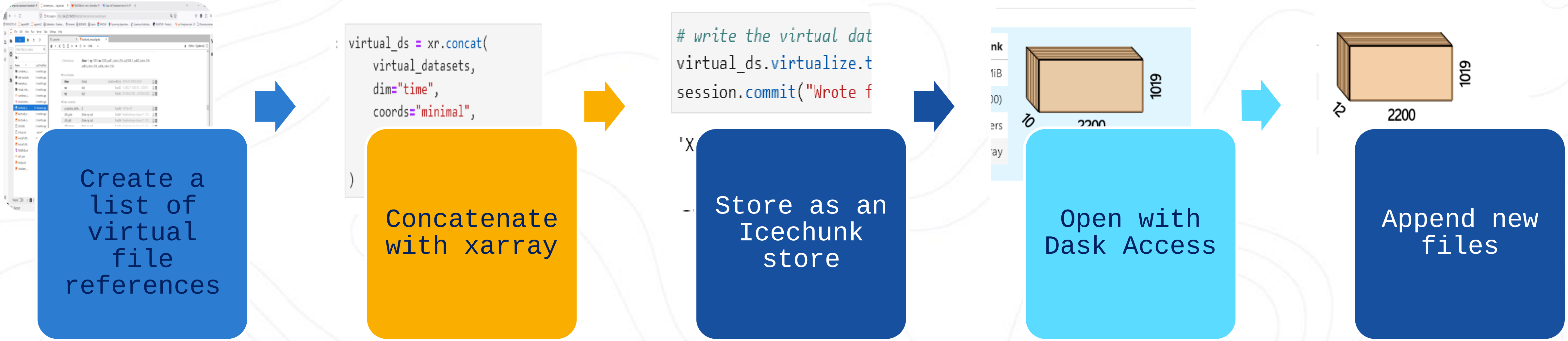
VICEPRESIDENCIA
TERCERA DEL GOBIERNO

MINISTERIO
PARA LA TRANSICIÓN ECOLÓGICA
Y EL RETO DEMOGRÁFICO

aemet
Agencia Estatal de Meteorología

EUMETSAT NWC SAF

icechunk workflow:



Create a list of virtual file references

Concatenate with xarray

```
virtual_ds = xr.concat(
    virtual_datasets,
    dim="time",
    coords="minimal",
)
```

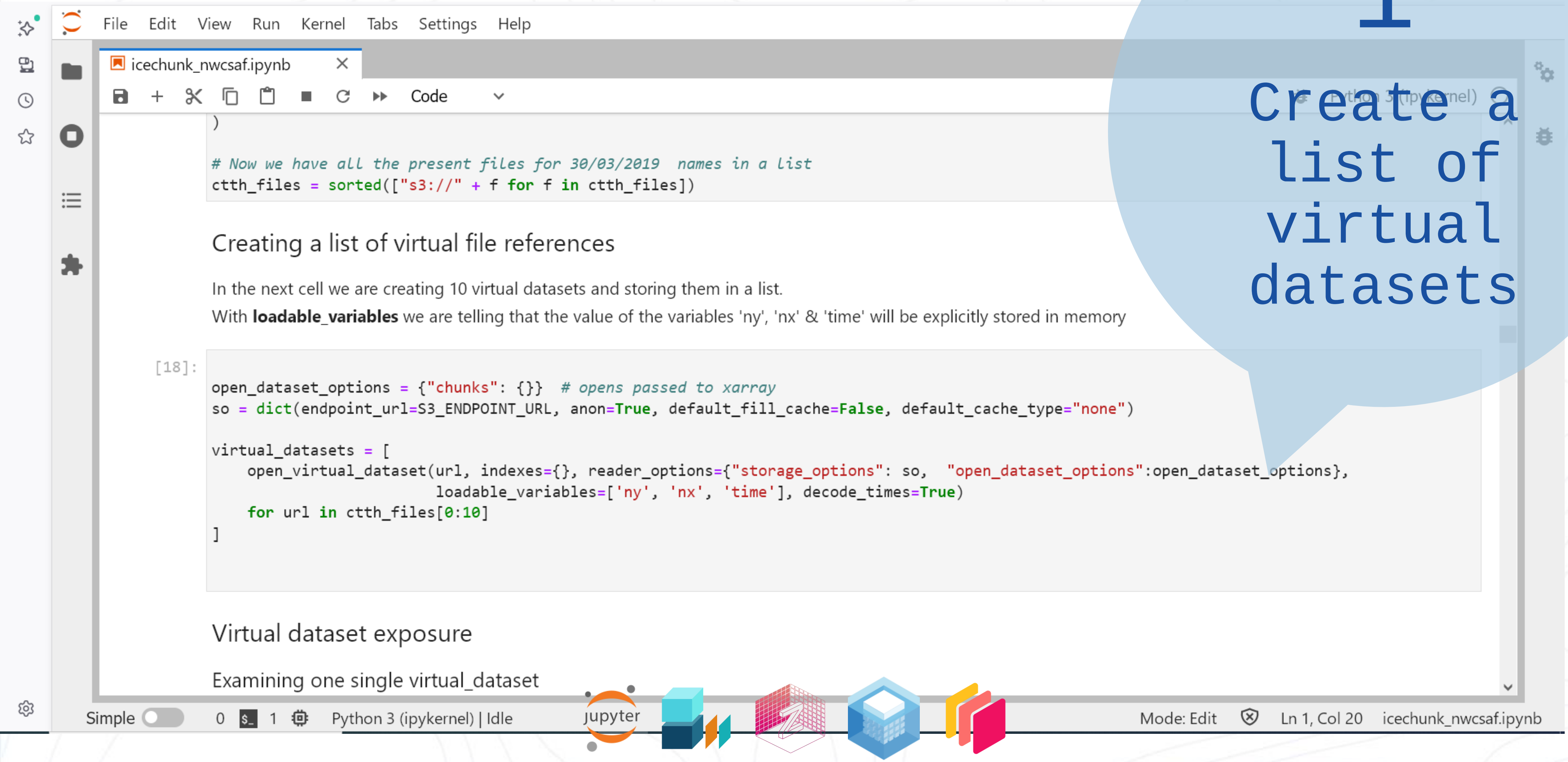
Store as an Icechunk store

```
# write the virtual dat
virtual_ds.virtualize.t
session.commit("Wrote f
```

Open with Dask Access

Append new files

1 Create a list of virtual datasets



```
)

# Now we have all the present files for 30/03/2019  names in a list
ctth_files = sorted(["s3://" + f for f in ctth_files])
```

Creating a list of virtual file references

In the next cell we are creating 10 virtual datasets and storing them in a list.
With **loadable_variables** we are telling that the value of the variables 'ny', 'nx' & 'time' will be explicitly stored in memory

[18]:

```
open_dataset_options = {"chunks": {}}  # opens passed to xarray
so = dict(endpoint_url=S3_ENDPOINT_URL, anon=True, default_fill_cache=False, default_cache_type="none")

virtual_datasets = [
    open_virtual_dataset(url, indexes={}, reader_options={"storage_options": so,  "open_dataset_options":open_dataset_options},
                         loadable_variables=['ny', 'nx', 'time'], decode_times=True)
    for url in ctth_files[0:10]
]
```

Virtual dataset exposure

Examining one single virtual_dataset

GOBIERNO DE ESPAÑA

VICEPRESIDENCIA TERCERA DEL GOBIERNO

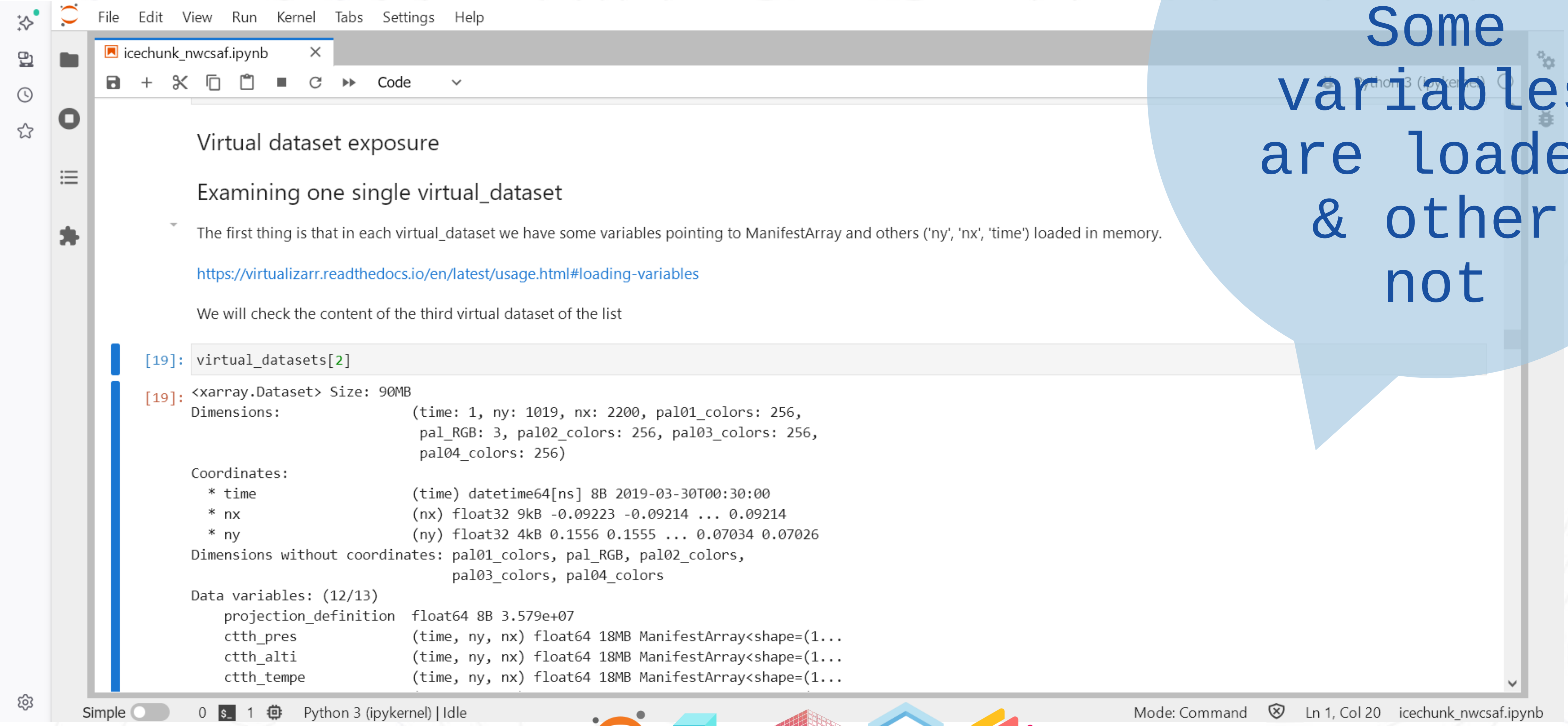
MINISTERIO PARA LA TRANSICIÓN ECOLÓGICA Y EL RETO DEMOGRÁFICO

aemet Agencia Estatal de Meteorología

EUMETSAT NWC SAF

2

Some variables are loaded & other not



Virtual dataset exposure

Examining one single virtual_dataset

The first thing is that in each virtual_dataset we have some variables pointing to ManifestArray and others ('ny', 'nx', 'time') loaded in memory.

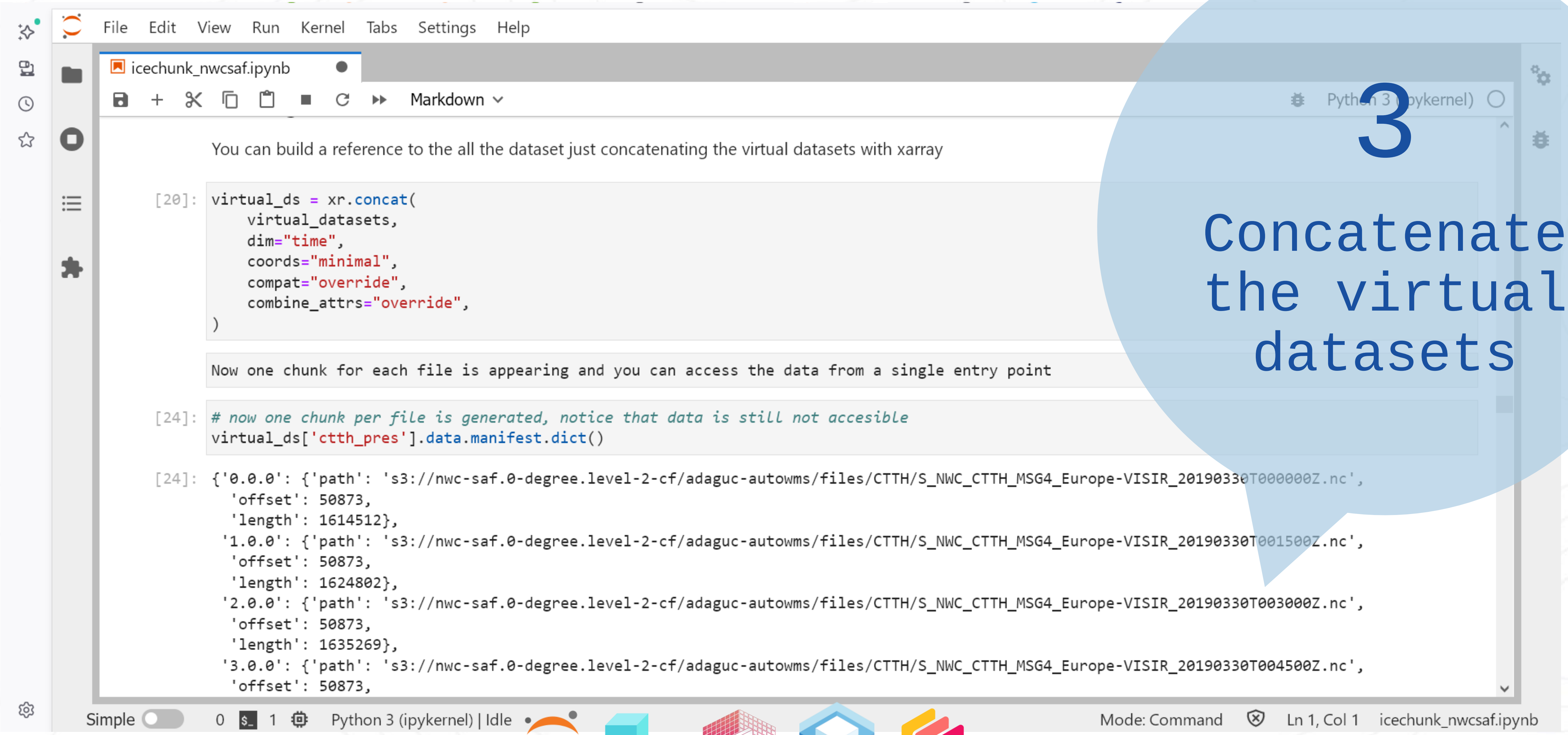
https://virtualizarr.readthedocs.io/en/latest/usage.html#loading-variables

We will check the content of the third virtual dataset of the list

```
[19]: virtual_datasets[2]
```

```
[19]: <xarray.Dataset> Size: 90MB
Dimensions:                 (time: 1, ny: 1019, nx: 2200, pal01_colors: 256,
                             pal_RGB: 3, pal02_colors: 256, pal03_colors: 256,
                             pal04_colors: 256)
Coordinates:
  * time                    (time) datetime64[ns] 8B 2019-03-30T00:30:00
  * nx                      (nx) float32 9kB -0.09223 -0.09214 ... 0.09214
  * ny                      (ny) float32 4kB 0.1556 0.1555 ... 0.07034 0.07026
Dimensions without coordinates: pal01_colors, pal_RGB, pal02_colors,
                                pal03_colors, pal04_colors
Data variables: (12/13)
    projection_definition  float64 8B 3.579e+07
    ctth_pres               (time, ny, nx) float64 18MB ManifestArray<shape=(1...
    ctth_alti               (time, ny, nx) float64 18MB ManifestArray<shape=(1...
    ctth_tempe              (time, ny, nx) float64 18MB ManifestArray<shape=(1...
```

jupyter



You can build a reference to the all the dataset just concatenating the virtual datasets with xarray

```
[20]: virtual_ds = xr.concat(
          virtual_datasets,
          dim="time",
          coords="minimal",
          compat="override",
          combine_attrs="override",
      )
```

Now one chunk for each file is appearing and you can access the data from a single entry point

```
[24]: # now one chunk per file is generated, notice that data is still not accesible
      virtual_ds['ctth_pres'].data.manifest.dict()
```

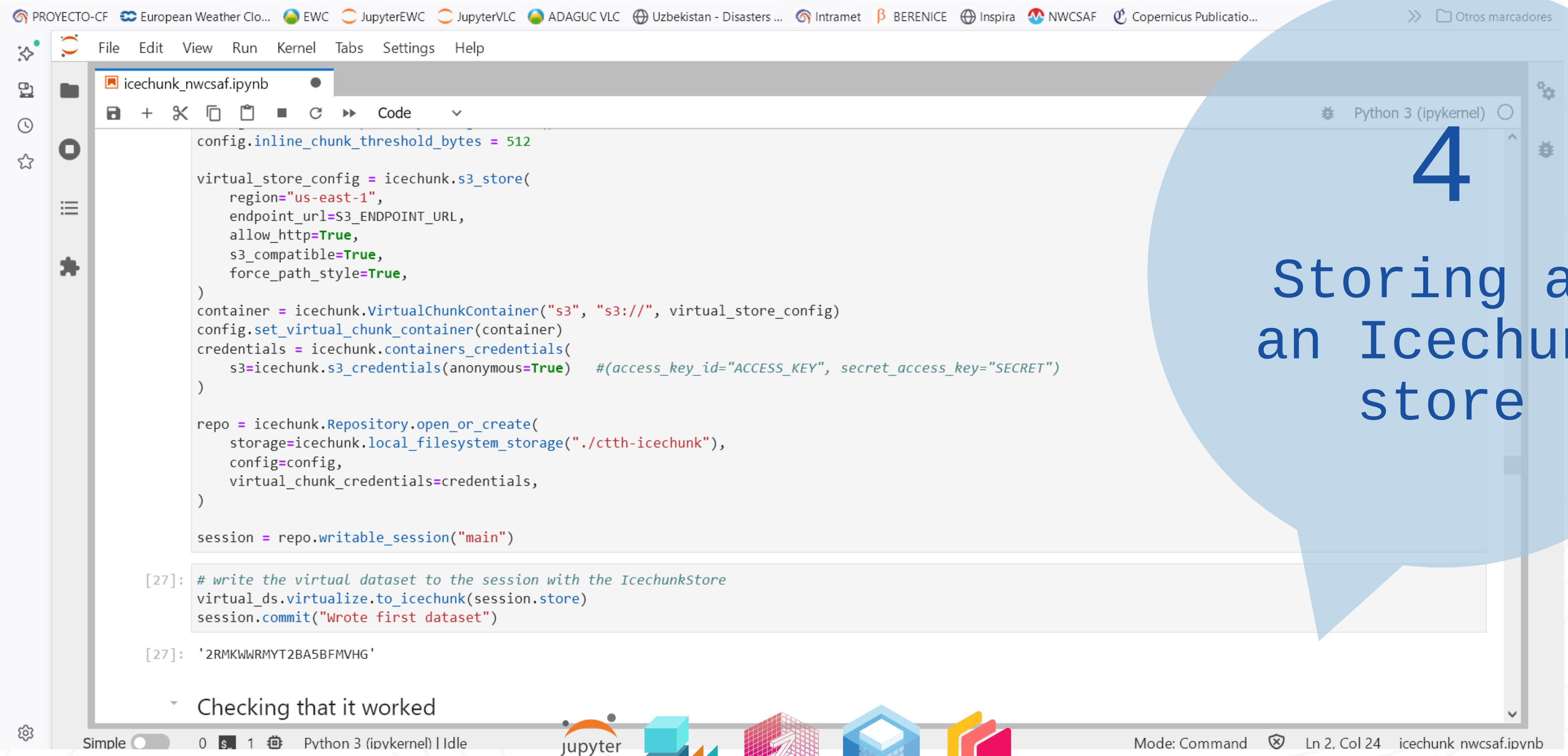
```
[24]: {'0.0.0': {'path': 's3://nwc-saf.0-degree.level-2-cf/adaguc-autowms/files/CTTH/S_NWC_CTTH_MSG4_Europe-VISIR_20190330T000000Z.nc',
        'offset': 50873,
        'length': 1614512},
       '1.0.0': {'path': 's3://nwc-saf.0-degree.level-2-cf/adaguc-autowms/files/CTTH/S_NWC_CTTH_MSG4_Europe-VISIR_20190330T001500Z.nc',
        'offset': 50873,
        'length': 1624802},
       '2.0.0': {'path': 's3://nwc-saf.0-degree.level-2-cf/adaguc-autowms/files/CTTH/S_NWC_CTTH_MSG4_Europe-VISIR_20190330T003000Z.nc',
        'offset': 50873,
        'length': 1635269},
       '3.0.0': {'path': 's3://nwc-saf.0-degree.level-2-cf/adaguc-autowms/files/CTTH/S_NWC_CTTH_MSG4_Europe-VISIR_20190330T004500Z.nc',
        'offset': 50873,
```

3

Concatenate the virtual datasets

4

Storing as an Icechunk store



```python
config.inline_chunk_threshold_bytes = 512

virtual_store_config = icechunk.s3_store(
    region="us-east-1",
    endpoint_url=S3_ENDPOINT_URL,
    allow_http=True,
    s3_compatible=True,
    force_path_style=True,
)
container = icechunk.VirtualChunkContainer("s3", "s3://", virtual_store_config)
config.set_virtual_chunk_container(container)
credentials = icechunk.containers_credentials(
    s3=icechunk.s3_credentials(anonymous=True)   #(access_key_id="ACCESS_KEY", secret_access_key="SECRET")
)

repo = icechunk.Repository.open_or_create(
    storage=icechunk.local_filesystem_storage("./ctth-icechunk"),
    config=config,
    virtual_chunk_credentials=credentials,
)

session = repo.writable_session("main")
```

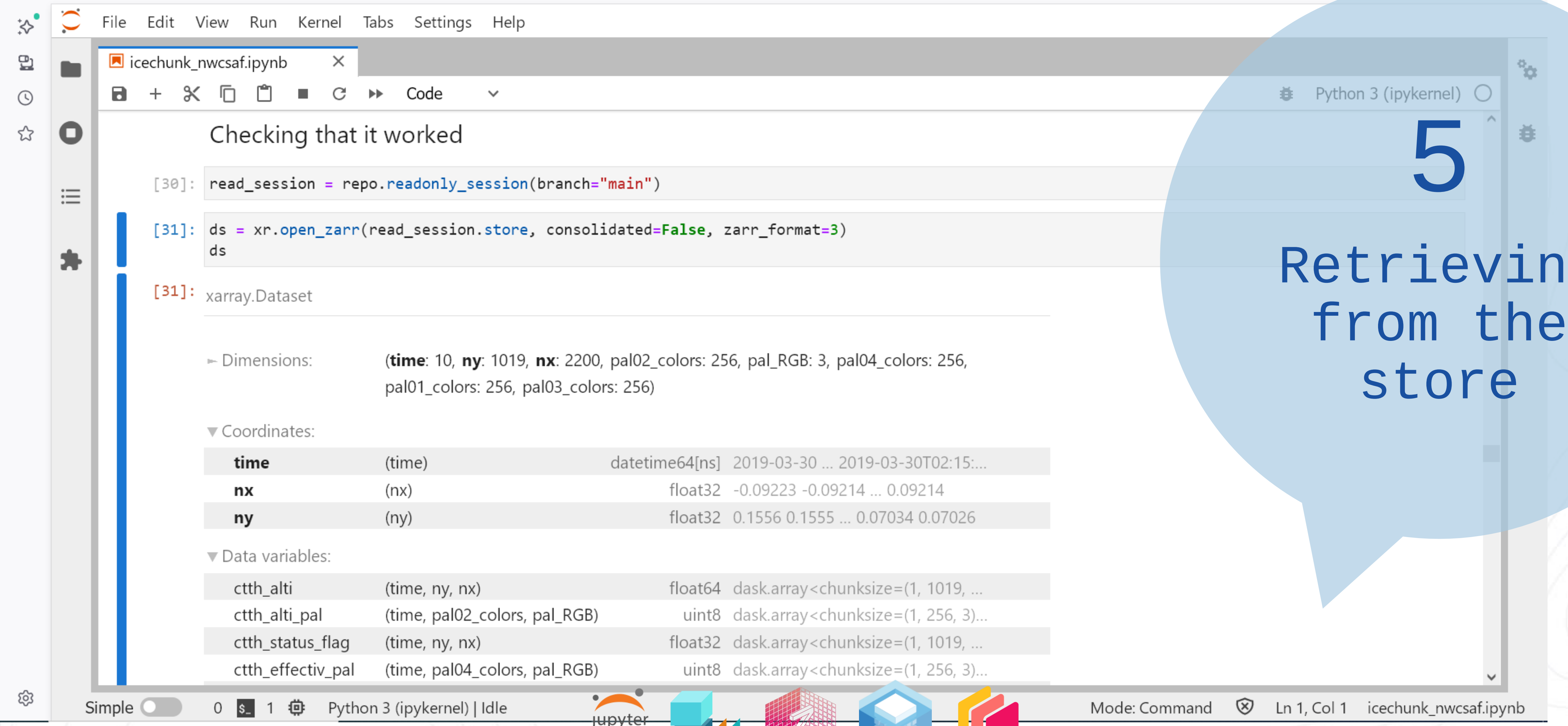
[27]:
```python
# write the virtual dataset to the session with the IcechunkStore
virtual_ds.virtualize.to_icechunk(session.store)
session.commit("Wrote first dataset")
```

[27]:
```
'2RMKWWRMYT2BA5BFMVHG'
```

Checking that it worked

5

Retrieving from the store



Checking that it worked

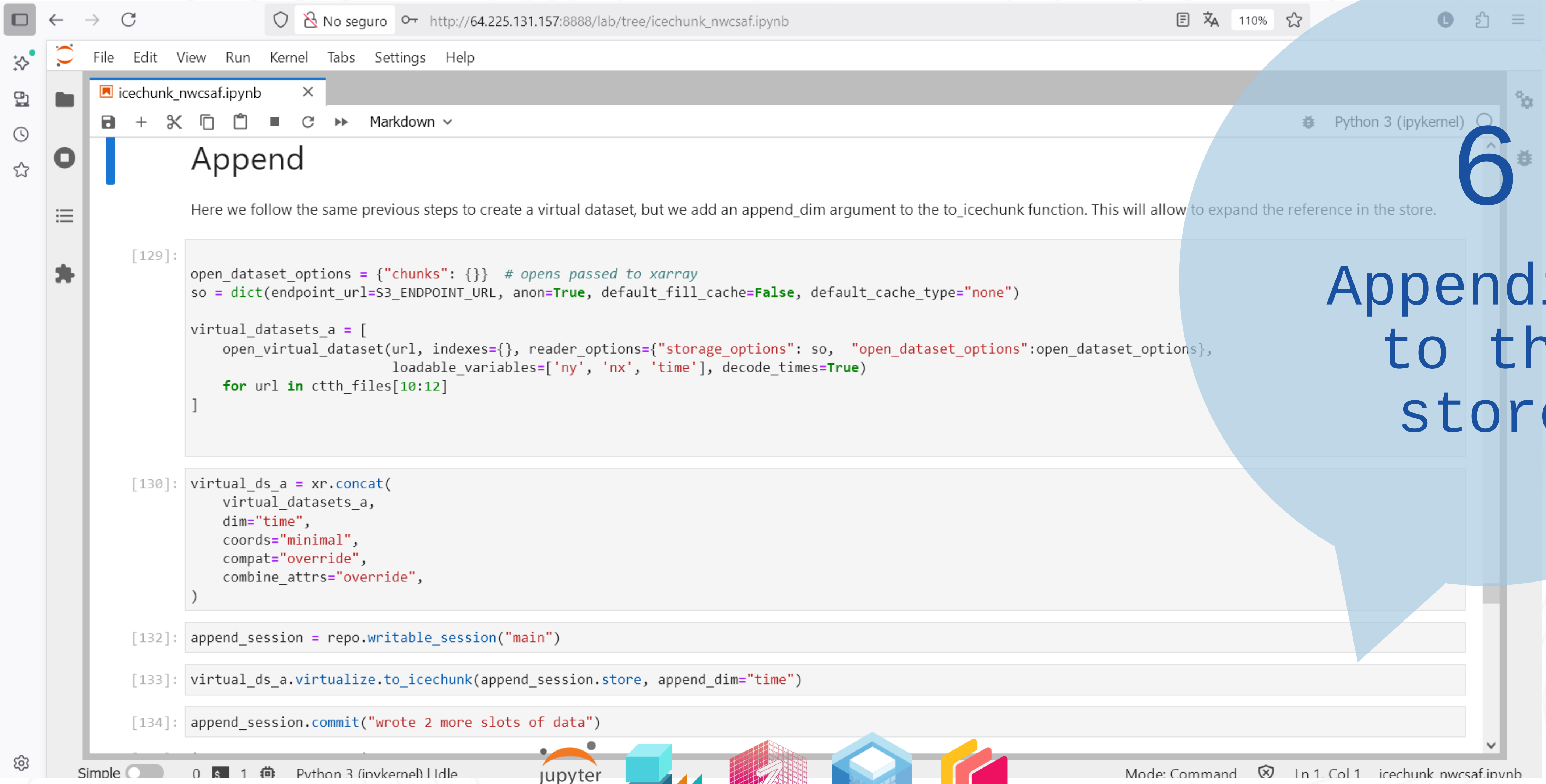
```
[30]: read_session = repo.readonly_session(branch="main")
```

```
[31]: ds = xr.open_zarr(read_session.store, consolidated=False, zarr_format=3)
      ds
```

```
[31]: xarray.Dataset

► Dimensions:        (time: 10, ny: 1019, nx: 2200, pal02_colors: 256, pal_RGB: 3, pal04_colors: 256,
                     pal01_colors: 256, pal03_colors: 256)

▼ Coordinates:
    time             (time)                          datetime64[ns]  2019-03-30 ... 2019-03-30T02:15:...
    nx               (nx)                                   float32  -0.09223 -0.09214 ... 0.09214
    ny               (ny)                                   float32  0.1556 0.1555 ... 0.07034 0.07026

▼ Data variables:
    ctth_alti        (time, ny, nx)                         float64  dask.array<chunksize=(1, 1019, ...
    ctth_alti_pal    (time, pal02_colors, pal_RGB)            uint8  dask.array<chunksize=(1, 256, 3)...
    ctth_status_flag (time, ny, nx)                         float32  dask.array<chunksize=(1, 1019, ...
    ctth_effectiv_pal (time, pal04_colors, pal_RGB)           uint8  dask.array<chunksize=(1, 256, 3)...
```

6

Appending to the store



Append

Here we follow the same previous steps to create a virtual dataset, but we add an append_dim argument to the to_icechunk function. This will allow to expand the reference in the store.

```
[129]:
open_dataset_options = {"chunks": {}}  # opens passed to xarray
so = dict(endpoint_url=S3_ENDPOINT_URL, anon=True, default_fill_cache=False, default_cache_type="none")

virtual_datasets_a = [
    open_virtual_dataset(url, indexes={}, reader_options={"storage_options": so,  "open_dataset_options":open_dataset_options},
                         loadable_variables=['ny', 'nx', 'time'], decode_times=True)
    for url in ctth_files[10:12]
]
```

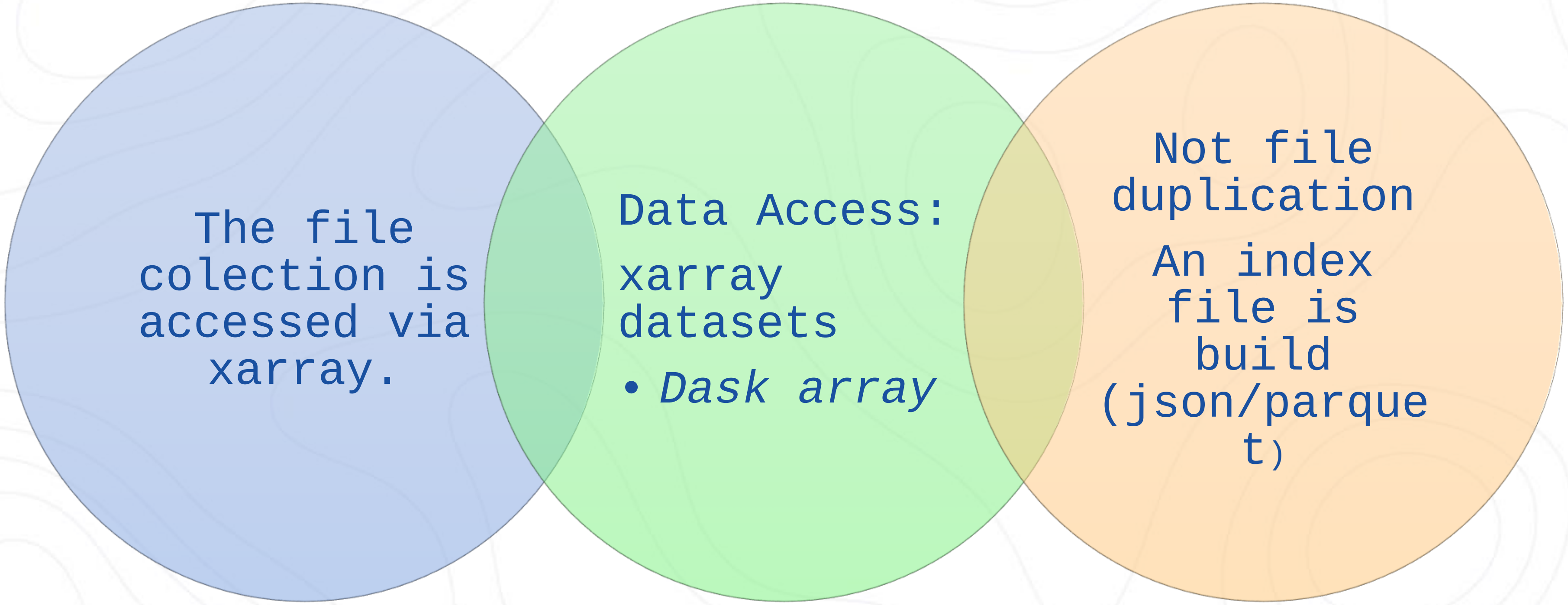
```
[130]:
virtual_ds_a = xr.concat(
    virtual_datasets_a,
    dim="time",
    coords="minimal",
    compat="override",
    combine_attrs="override",
)
```

```
[132]: append_session = repo.writable_session("main")
```

```
[133]: virtual_ds_a.virtualize.to_icechunk(append_session.store, append_dim="time")
```

```
[134]: append_session.commit("wrote 2 more slots of data")
```

Features



The file colection is accessed via xarray.

Data Access:

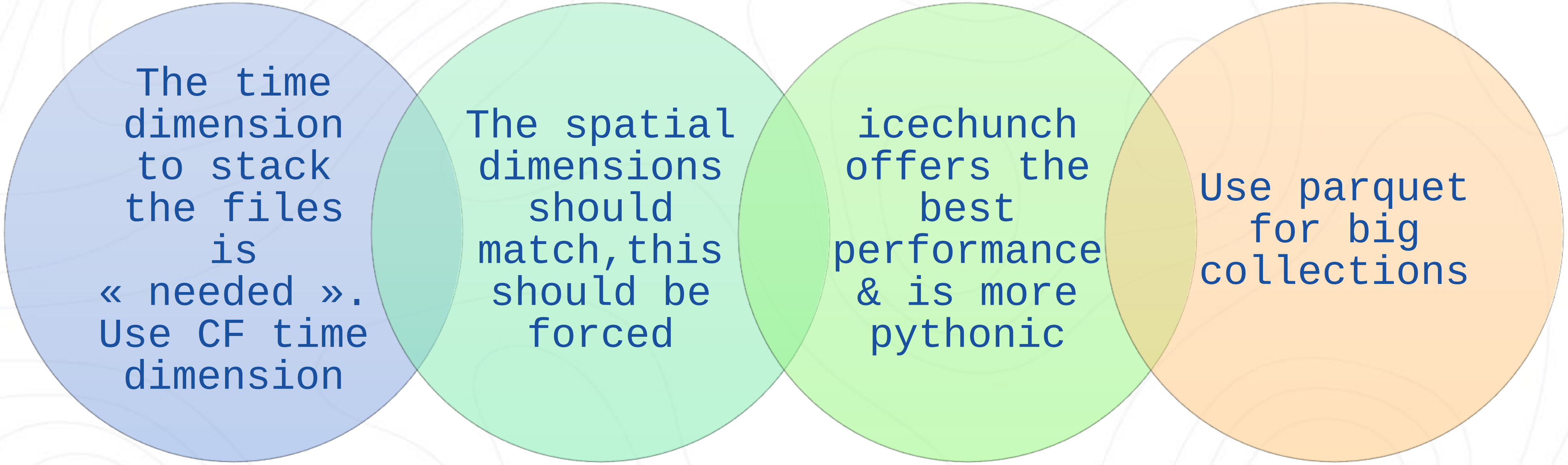
xarray datasets

- *Dask array*

Not file duplication

An index file is build (json/parquet)

Lessons learned

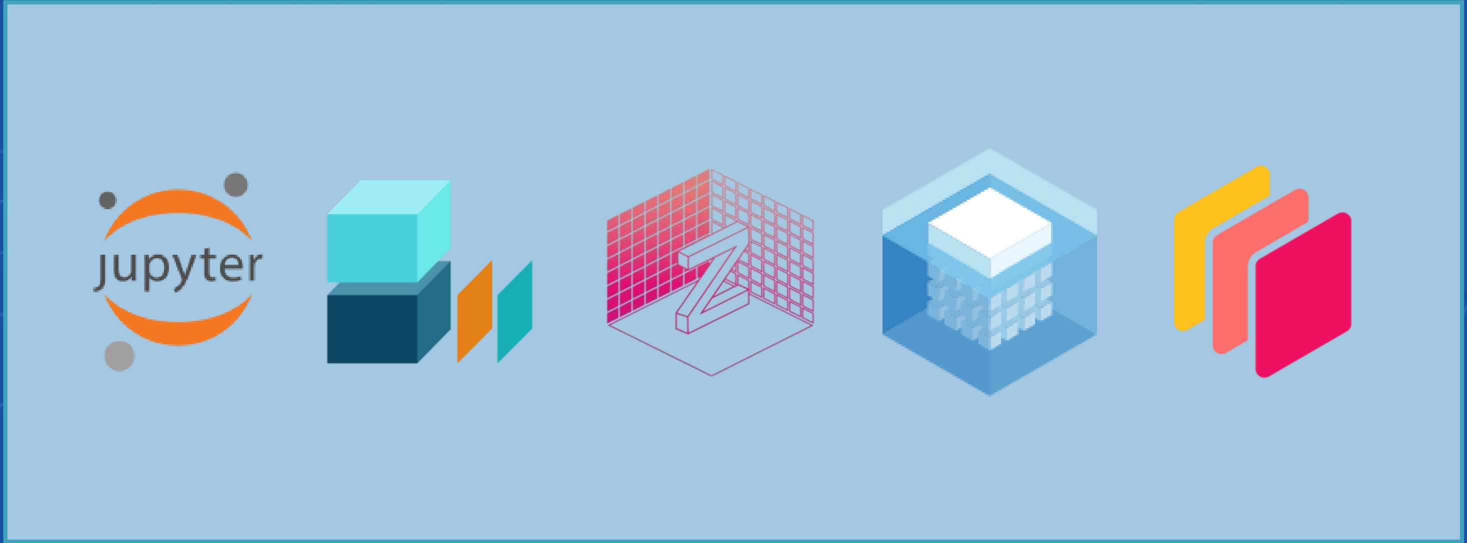


The time dimension to stack the files is « needed ». Use CF time dimension

The spatial dimensions should match, this should be forced

icechunch offers the best performance & is more pythonic

Use parquet for big collections



jllisov@aemet.es

GOBIERNO
DE ESPAÑA

VICEPRESIDENCIA
TERCERA DEL GOBIERNO

MINISTERIO
PARA LA TRANSICIÓN ECOLÓGICA
Y EL RETO DEMOGRÁFICO

aemet
Agencia Estatal de Meteorología

EUMETSAT
NWC SAF