EUMETSAT Short Courses: Weather-related satellite data with Jupyter Notebooks

Natasa Strelec Mahovic, Alen Berta and Ivan Smiljanic

09 November 2021



Agenda

Jupyter Notebooks – a short introduction

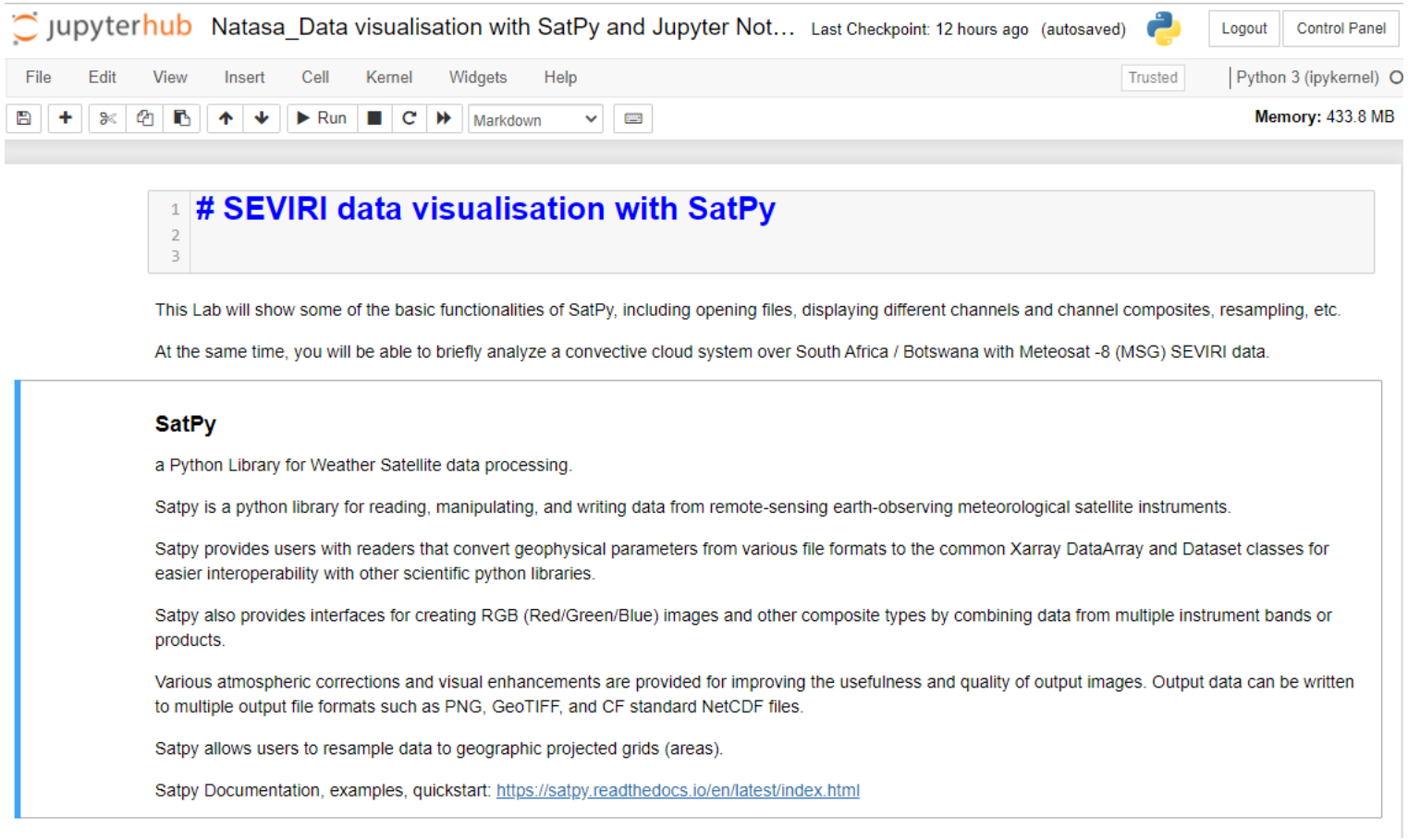
SEVIRI data visualization with SatPy

Severe convection - SEVIRI, MODIS, CloudSat and VIIRS

You will see how you can:

- Load and display satellite data using Jupyter Notebooks
- Visualize single channels and channel combinations (RGB products)
- Analyse cloud types and basic weather patterns in visualized images

SEVIRI data visualization with Satpy



jupyterhub Natasa_Data visualisation with SatPy and Jupyter Not… Last Checkpoint: 12 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code **Memory:** 433.9 MB

How to import data?

For this Notebook, we use SEVIRI data in NATIVE format. The file, MSG3-SEVI-MSG15-0100-NA-20151106141242.255000000Z-20151106141259-1504186.nat, contains SEVIRI data in all 12 channels for 6 November 2015.

First import needed libraries, location of the image/scene (make sure that you are using backslash) and load the scene with the respective reader.

In [ ]:

```
from satpy.scene import Scene
from satpy.resample import get_area_def
from satpy import find_files_and_readers
from datetime import datetime
import glob
import warnings
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')


# Date format in YYYYMMDDhhmm. Change the link to the corresponding folder. Make sure to change "\" (backslash) into forward
fnames = glob.glob('/data/Weather/Summer-School-Bracciano_13-17_Sep_2021/Data/JN_data/JN_lab1_introduction/seviri/*201511061

scn = Scene(reader='seviri_l1b_native',  filenames=fnames)


```

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 12 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown Memory: 422.8 MB

List available readers for more information.

In [2]:
```
from satpy import available_readers
available_readers()
```

Out[2]:
```
['abi_l1b',
 'abi_l1b_scmi',
 'abi_l2_nc',
 'agri_l1',
 'ahi_hrit',
 'ahi_hsd',
 'ahi_l1b_gridded_bin',
 'amsr2_l1b',
 'amsr2_l2',
 'amsr2_l2_gaasp',
 'avhrr_l1b_aapp',
 'avhrr_l1b_eps',
 'avhrr_l1b_hrpt',
 'avhrr_l1c_eum_gac_fdr_nc',
 'electrol_hrit',
 'fci_l2_nc',
 'glm_l2',
 'goes-imager_hrit',
 'goes-imager_nc',
 'gpm_imerg',
 'hy2_scat_l2b_h5',
 'iasi_l2',
 'jami_hrit',
 'maia',
 'mirs',
 'modis_l1b',
 'modis_l2',
 'mtsat2-imager_hrit',
 'mviri_l1b_fiduceo_nc',
 'nwcsaf-geo',
 'nwcsaf-msg2013-hdf5',
 'nwcsaf-pps_nc',
 'olci_l1b',
 'olci_l2',
 'omps_edr',
 'safe_sar_l2_ocn',
 'satpy_cf_nc',
 'seviri_l1b_hrit',
 'seviri_l1b_icare',
 'seviri_l1b_native',
 'seviri_l1b_nc',
 'slstr_l1b',
 'slstr_l2',
 'vaisala_gld360',
```

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 12 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code Memory: 2.6 GB

```
'seviri_l1b_native',
'seviri_l1b_nc',
'slstr_l1b',
'slstr_l2',
'vaisala_gld360',
'viirs_compact',
'viirs_edr_flood',
'viirs_sdr']
```

List available dataset/bands from the loaded scene

In [3]:
```
1 scn.available_dataset_names()
```

Out[3]:
```
['HRV',
 'IR_016',
 'IR_039',
 'IR_087',
 'IR_097',
 'IR_108',
 'IR_120',
 'IR_134',
 'VIS006',
 'VIS008',
 'WV_062',
 'WV_073']
```

Load wanted dataset and show its atributes/keys. You can also load the dataset by entering the specific wavelenght- in this case ([10.8])

In [ ]:
```
1 scn.load(['IR_108'], upper_right_corner='NE')
2
3 scn.keys()
```

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 12 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown **Memory:** 4.3 GB

In [4]:
```
scn.load(['IR_108'], upper_right_corner='NE')

scn.keys()
```

```
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name upsidedown.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name leftright.
```

Out[4]:
```
[DataID(name='IR_108', wavelength=WavelengthRange(min=9.8, central=10.8, max=11.8, unit='µm'), resolution=3000.403165817, calib
ration=<calibration.brightness_temperature>, modifiers=())]
```

You can check the area atributes with the following code.

In [5]:
```
scn['IR_108'].attrs['area']
```

Out[5]:
```
Area ID: msg_seviri_fes_3km
Description: MSG SEVIRI Full Earth Scanning service area definition with 3 km resolution
Projection: {'a': '6378169', 'h': '35785831', 'lon_0': '0', 'no_defs': 'None', 'proj': 'geos', 'rf': '295.488065897014', 'typ
e': 'crs', 'units': 'm', 'x_0': '0', 'y_0': '0'}
Number of columns: 3712
Number of rows: 3712
Area extent: (-5568748.2758, -5568748.2758, 5568748.2758, 5568748.2758)
```

Or you can print the metadata of the loaded image.

In [6]:
```
print(scn)
```

```
<xarray.DataArray 'reshape-c30c2dcdc3017228e42635090d93974c' (y: 3712, x: 3712)>
dask.array<getitem, shape=(3712, 3712), dtype=float32, chunksize=(3712, 3712), chunktype=numpy.ndarray>
Coordinates:
    acq_time  (y) datetime64[ns] NaT NaT NaT NaT NaT NaT ... NaT NaT NaT NaT NaT
    crs       object PROJCRS["unknown",BASEGEOGCRS["unknown",DATUM["unknown",...
  * y         (y) float64 5.567e+06 5.564e+06 ... -5.564e+06 -5.567e+06
  * x         (x) float64 -5.567e+06 -5.564e+06 ... 5.564e+06 5.567e+06
Attributes: (12/17)
    orbital_parameters:       {'projection_longitude': 0.0, 'projection_latit...
    units:                    K
    wavelength:               10.8 µm (9.8-11.8 µm)
    standard_name:            toa_brightness_temperature
    platform_name:            Meteosat-10
    sensor:                   seviri
    ...                       ...
    name:                     IR_108
    resolution:               3000.403165817
    calibration:              brightness_temperature
    modifiers:                ()
    _satpy_id:                DataID(name='IR_108', wavelength=WavelengthRang...
    ancillary_variables:      []
```

jupyterhub Natasa_Data visualisation with SatPy and Jupyter Not… Last Checkpoint: 12 hours ago (autosaved) Logout Control Panel

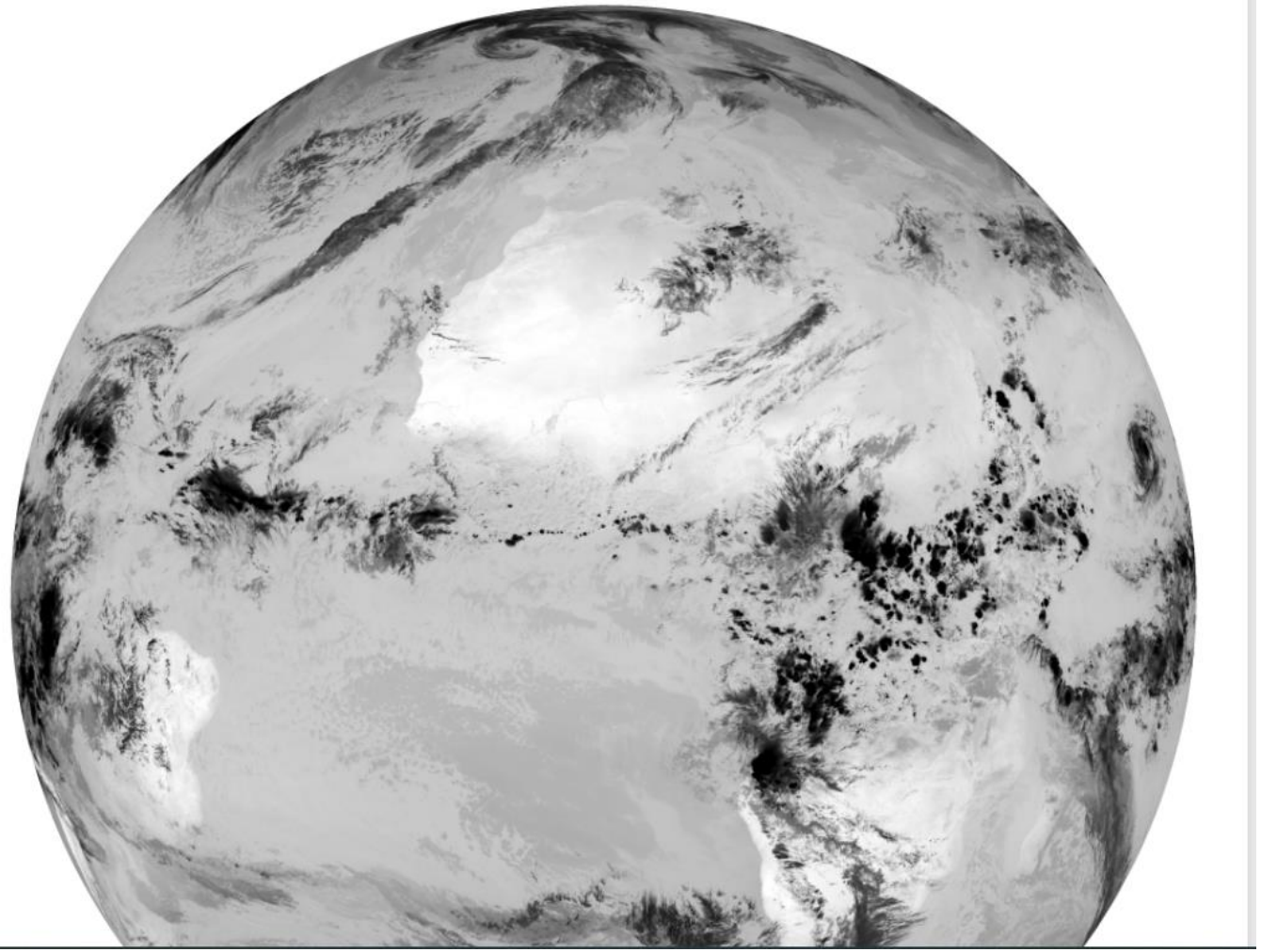
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

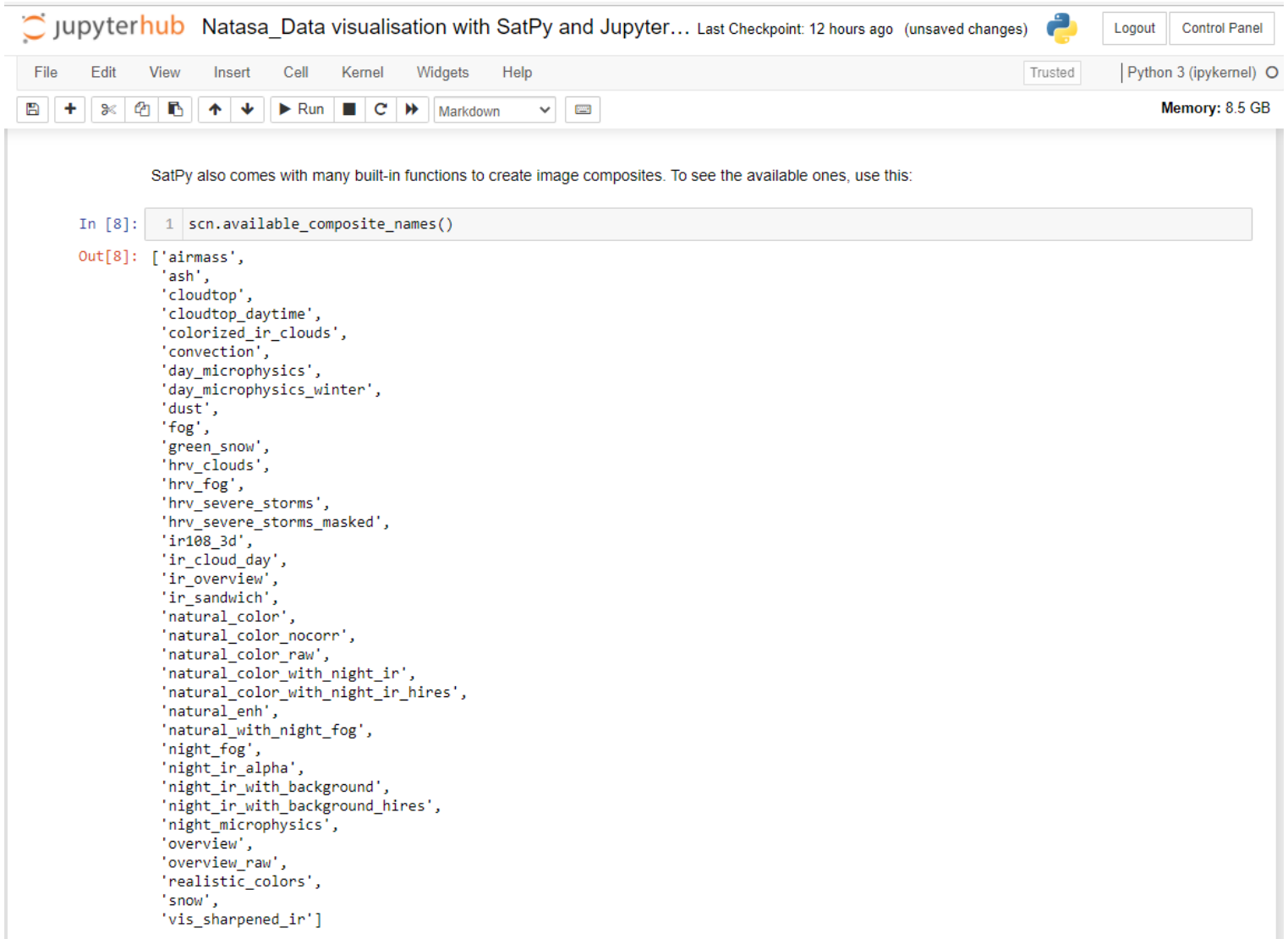
Run Markdown Memory: 7 GB

Now show the loaded band.

In [7]:
```
1 scn.show('IR_108')
```

Out[7]:




jupyterhub
Natasa_Data visualisation with SatPy and Jupyter…
Last Checkpoint: 12 hours ago (unsaved changes)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
Memory: 8.5 GB
SatPy also comes with many built-in functions to create image composites. To see the available ones, use this:
In [8]:
1 scn.available_composite_names()
Out[8]:
['airmass',
'ash',
'cloudtop',
'cloudtop_daytime',
'colorized_ir_clouds',
'convection',
'day_microphysics',
'day_microphysics_winter',
'dust',
'fog',
'green_snow',
'hrv_clouds',
'hrv_fog',
'hrv_severe_storms',
'hrv_severe_storms_masked',
'ir108_3d',
'ir_cloud_day',
'ir_overview',
'ir_sandwich',
'natural_color',
'natural_color_nocorr',
'natural_color_raw',
'natural_color_with_night_ir',
'natural_color_with_night_ir_hires',
'natural_enh',
'natural_with_night_fog',
'night_fog',
'night_ir_alpha',
'night_ir_with_background',
'night_ir_with_background_hires',
'night_microphysics',
'overview',
'overview_raw',
'realistic_colors',
'snow',
'vis_sharpened_ir']

jupyterhub Natasa_Data visualisation with SatPy and Jupyter Not… Last Checkpoint: 12 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

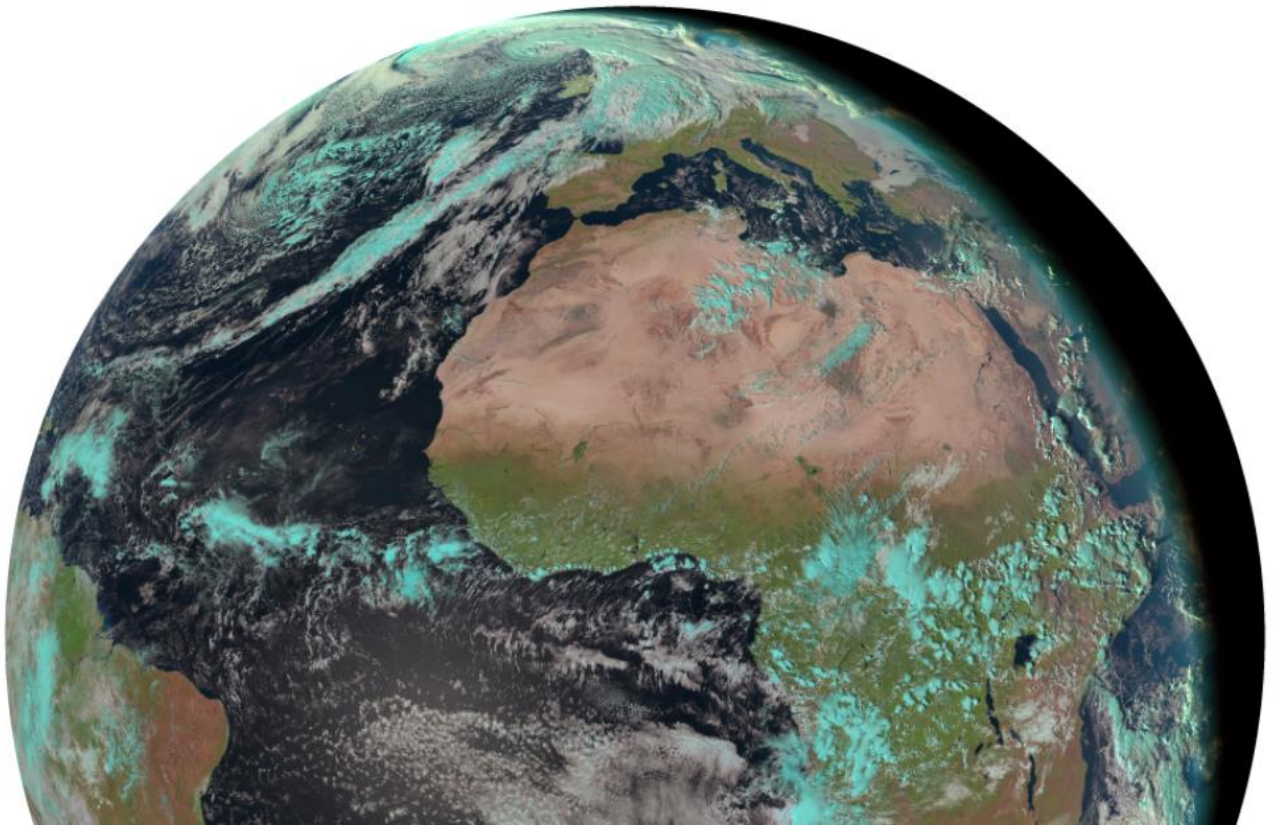
Run Code Memory: 9.1 GB

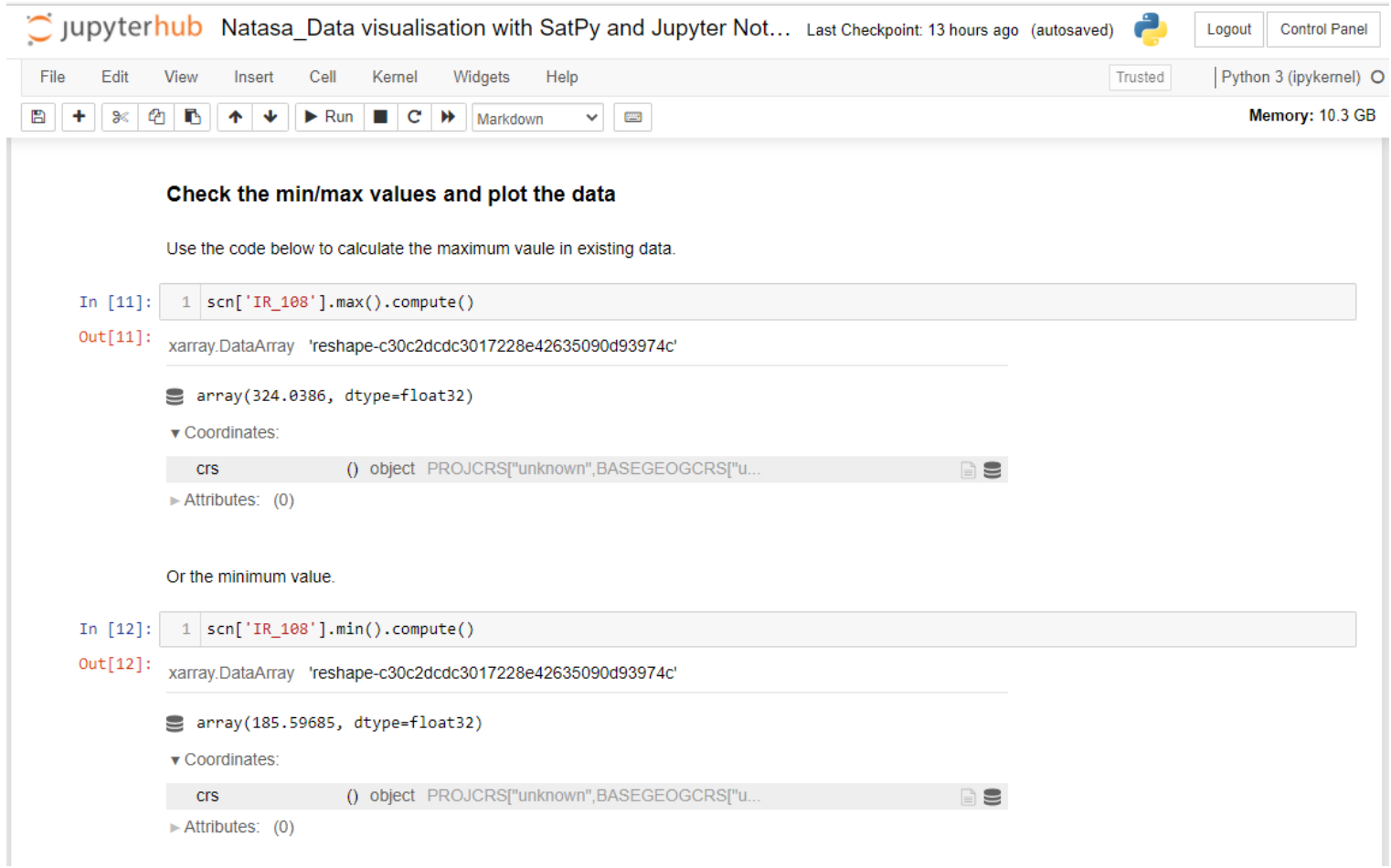
As with the single bands, you have to load a composite into the Scene object first, if you want to work with it:

In [9]:
```
1 scn.load(["natural_color"], upper_right_corner='NE')
2 scn.show("natural_color")
```

```
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name upsidedown.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name leftright.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name upsidedown.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name leftright.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name upsidedown.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name leftright.
```

Out[9]:




jupyterhub
Natasa_Data visualisation with SatPy and Jupyter Not…
Last Checkpoint: 13 hours ago (autosaved)
Logout
Control Panel
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Markdown
Memory: 10.3 GB
Check the min/max values and plot the data
Use the code below to calculate the maximum vaule in existing data.
In [11]:
1 scn['IR_108'].max().compute()
Out[11]:
xarray.DataArray 'reshape-c30c2dcdc3017228e42635090d93974c'
array(324.0386, dtype=float32)
Coordinates:
crs () object PROJCRS["unknown",BASEGEOGCRS["u...
Attributes: (0)
Or the minimum value.
In [12]:
1 scn['IR_108'].min().compute()
Out[12]:
xarray.DataArray 'reshape-c30c2dcdc3017228e42635090d93974c'
array(185.59685, dtype=float32)
Coordinates:
crs () object PROJCRS["unknown",BASEGEOGCRS["u...
Attributes: (0)

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown Memory: 10.4 GB

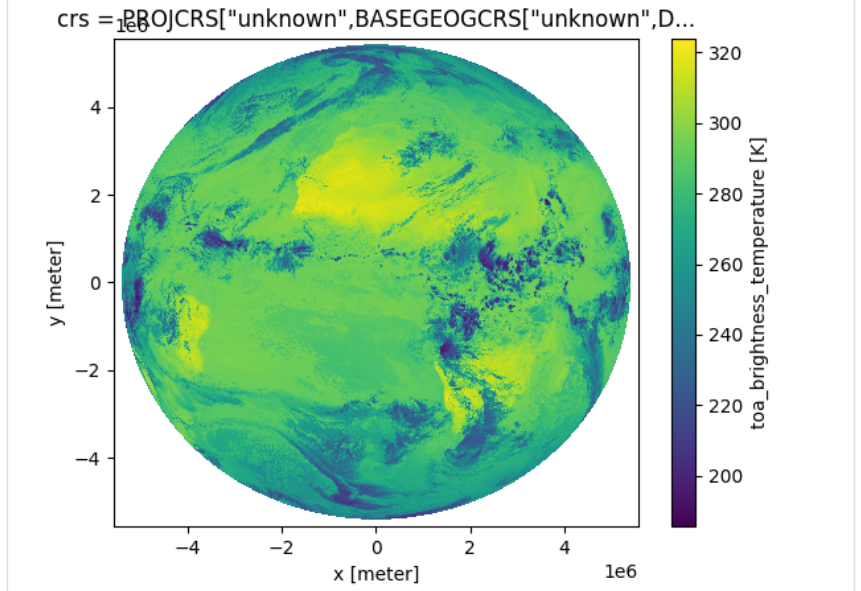
Display an interactive plot

```
show() function does not allow to print the data with colorbar ie. to get an impresion of the value range.

Now lets plot the image in the interactive plot with matplotlib in order to explore it better. Below loaded plot you have commands to zoom, pan, move. **Also when you hover the mouse over the plot, in lower right corner you can see the value.**
Download is not working but at this point we do not need it.
```

In [13]:

```
#this part of code make the plot interactive
%matplotlib notebook
scn['IR_108'].plot.imshow()
```

Figure 1



Out[13]: <matplotlib.image.AxesImage at 0x7f9578909c50>

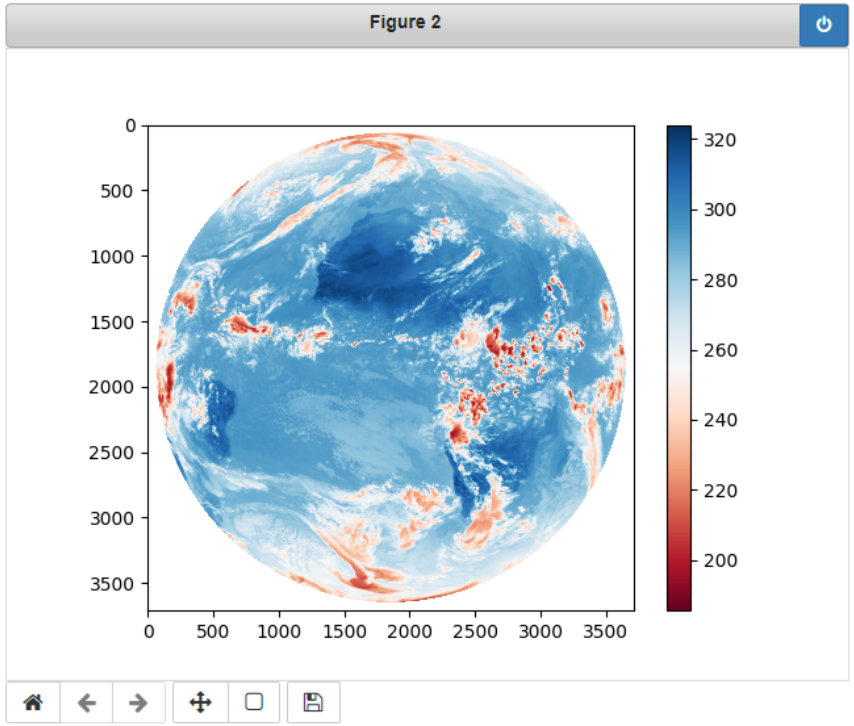
jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code **Memory:** 10.9 GB

But this is a default view and colorbar. To add a custom colormaps use the code below. Change the cmap to "RdBu". Try it. More color maps can be found here: https://matplotlib.org/stable/tutorials/colors/colormaps.html

In [14]:
```
1 plt.figure()
2 plt.imshow(scn['IR_108'].values, cmap="RdBu")
3 plt.colorbar()
```



Out[14]: <matplotlib.colorbar.Colorbar at 0x7f9571075cc0>

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown Memory: 11.6 GB

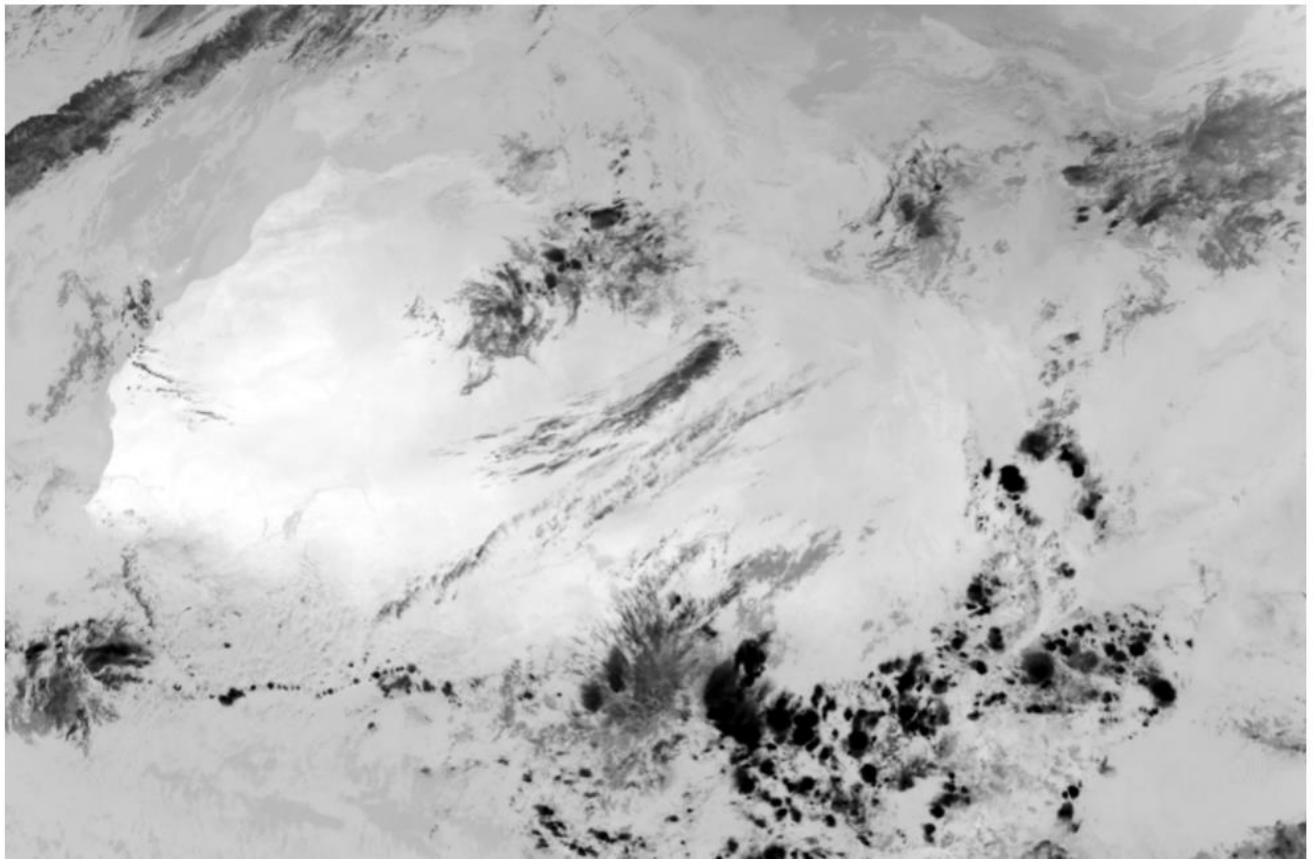
Resample

Now, lets resample to a smaller region-Africa. Predefined areas are located here https://github.com/pytroll/satpy/blob/main/satpy/etc/areas.yaml. For creating a new area, please refer to the SatPy documentation.

In [15]:
```
1 local_scene = scn.resample("africa")
2 local_scene.show('IR_108')
3
```

```
INFO:satpy.resample:Using default KDTree resampler
```

Out[15]:



jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

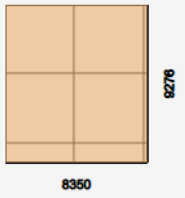
Run Markdown Memory: 12 GB

Check the metadata

In [16]:
```
1 local_scene['IR_108']
```

Out[16]: xarray.DataArray 'my_index-cf14d93119c76e2673ac250c91b95020' (**y**: 9276, **x**: 8350)

	Array	Chunk
Bytes	295.47 MiB	64.00 MiB
Shape	(9276, 8350)	(4096, 4096)
Count	278 Tasks	9 Chunks
Type	float32	numpy.ndarray



▾ Coordinates:

crs	()	object	PROJCRS["unknown",BASEGEOGCRS["u...
y	(y)	float64	4.48e+06 4.479e+06 ... -4.795e+06
x	(x)	float64	-4.458e+06 -4.457e+06 ... 3.891e+06

Check again for the minimum value. **Note it**.

In [17]:
```
1 local_scene['IR_108'].min().compute()
```

Out[17]: xarray.DataArray 'my_index-cf14d93119c76e2673ac250c91b95020'

```
array(185.59685, dtype=float32)
```

▾ Coordinates:

crs	()	object	PROJCRS["unknown",BASEGEOGCRS["u...

▸ Attributes: (0)

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

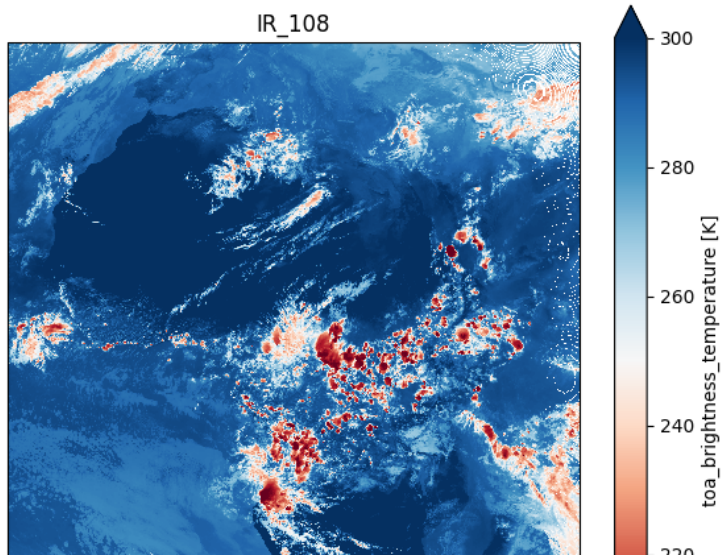
Run Markdown Memory: 12.3 GB

```
1 Now plot again just resampled data. You can also set the size of the figure (figsize) or change the colormap (cmap) or
  change opacity with alpha or range with vmin/vmax.
2
3 For instance, to see the coldest areas change vmin and vmax to a range from 185 - 210 K and run again this cell.
4
5 <mark>**Try to find the coldest pixel- minimum value in the plot.**</mark>
```

In [18]:

```
1 plt.figure(figsize=(7, 7))
2
3 crs =local_scene['IR_108'].attrs['area'].to_cartopy_crs()
4 ax = plt.axes(projection=crs)
5
6 local_scene['IR_108'].plot.imshow(cmap="RdBu", vmin=200, vmax=300, alpha=1) #, vmin=185, vmax=210)
7 plt.title("IR_108")
8 plt.show()
9
```

Figure 3

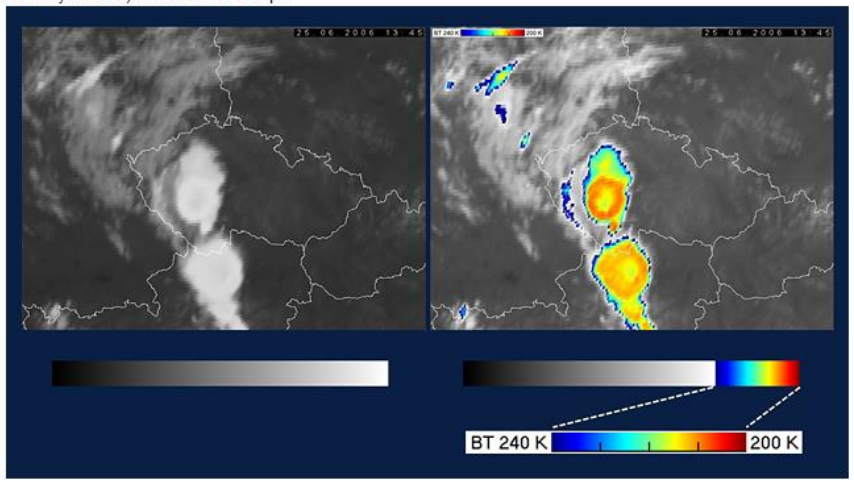


jupyterhub Natasa_Data visualisation with SatPy and Jupyter Not… Last Checkpoint: 13 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code Memory: 12.3 GB

You probably heard of (or at least you saw) Setvak color map.



"The enhancement of grey or color scales, used for visualization of thermal IR-window imagery, has been utilized since the early days of weather satellites. The purpose of this technique is obvious – to provide a human eye more details in a specific range of temperatures, details which otherwise would remain hidden." More info here: https://cwg.eumetsat.int/color-enhancements/

Let´s create this colormap

In [ ]:

```
# Create new colormap- SETVAK
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.colors import ListedColormap, LinearSegmentedColormap

bottom = cm.get_cmap("Greys", 98)
top = cm.get_cmap('jet_r', 148)

newcolors = np.vstack((top(np.linspace(0, 1, 98)),
                       bottom(np.linspace(0, 1, 148))))
newcmp = ListedColormap(newcolors, name='Setvak')

```

jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

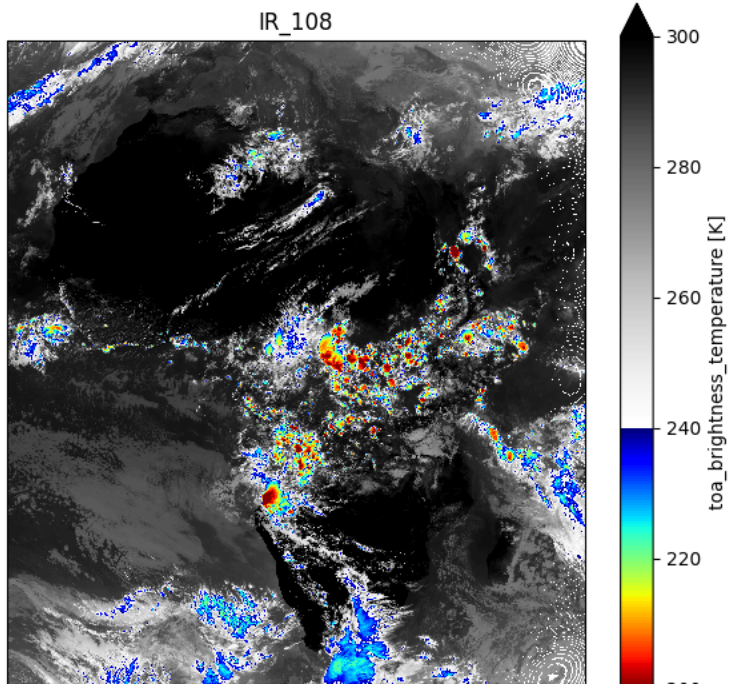
Run Code **Memory:** 12.7 GB

```
1 Now plot again with new colormap.
```

In [20]:

```
1  plt.figure(figsize=(7, 7))
2
3
4  crs =local_scene['IR_108'].attrs['area'].to_cartopy_crs()
5  ax = plt.axes(projection=crs)
6
7  local_scene['IR_108'].plot.imshow(cmap=newcmp, vmin=200, vmax=300, alpha=1)
8  plt.title("IR_108")
9  plt.show()
10
```

Figure 4



jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 13 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown Memory: 12.9 GB

Display the SEVIRI HRV Image and compare to IR10.8 image

SEVIRI have 3 different calibrations per band- brightness_temperature, reflectance and radiance.

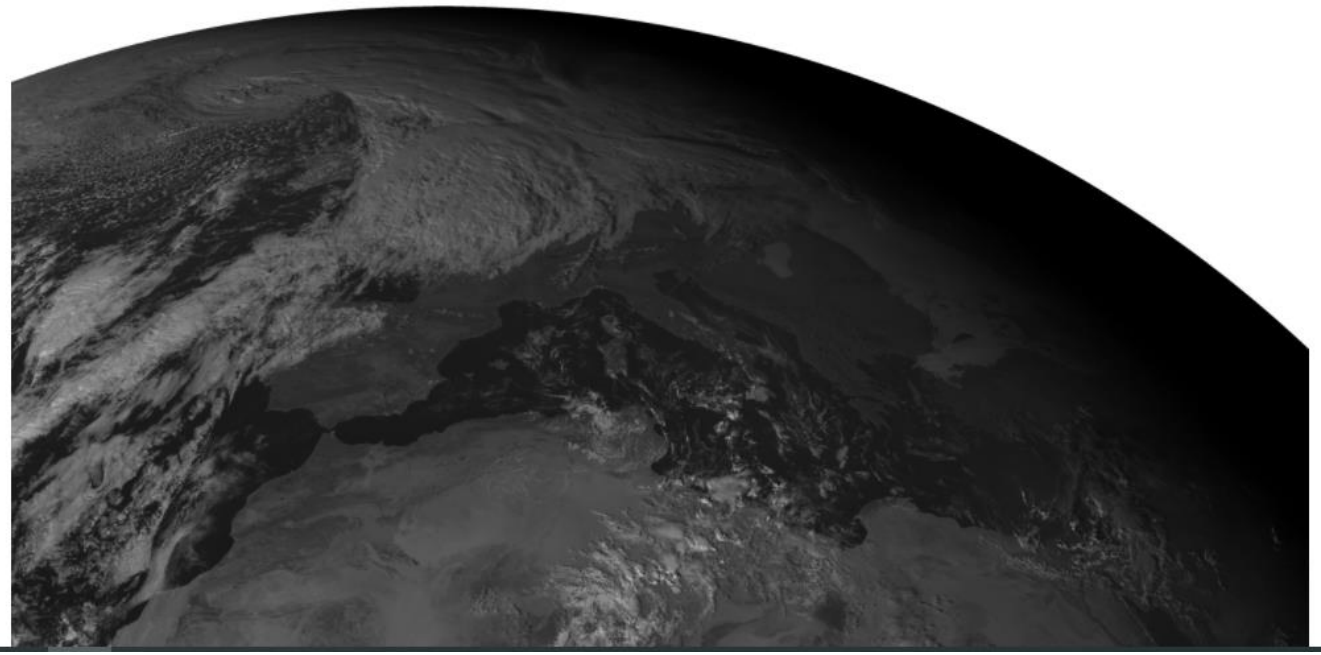
Load HRV band and only reflectance.

In [21]:

```
1 scn.load(['HRV'], calibration=["reflectance"], upper_right_corner='NE')
2
3 print(scn.attrs['end_time'])
4
5 scn.show('HRV')
```

```
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name upsidedown.
INFO:satpy.readers.yaml_reader:Flipping Dataset unknown_name leftright.
2015-11-06 14:15:10.122358
```

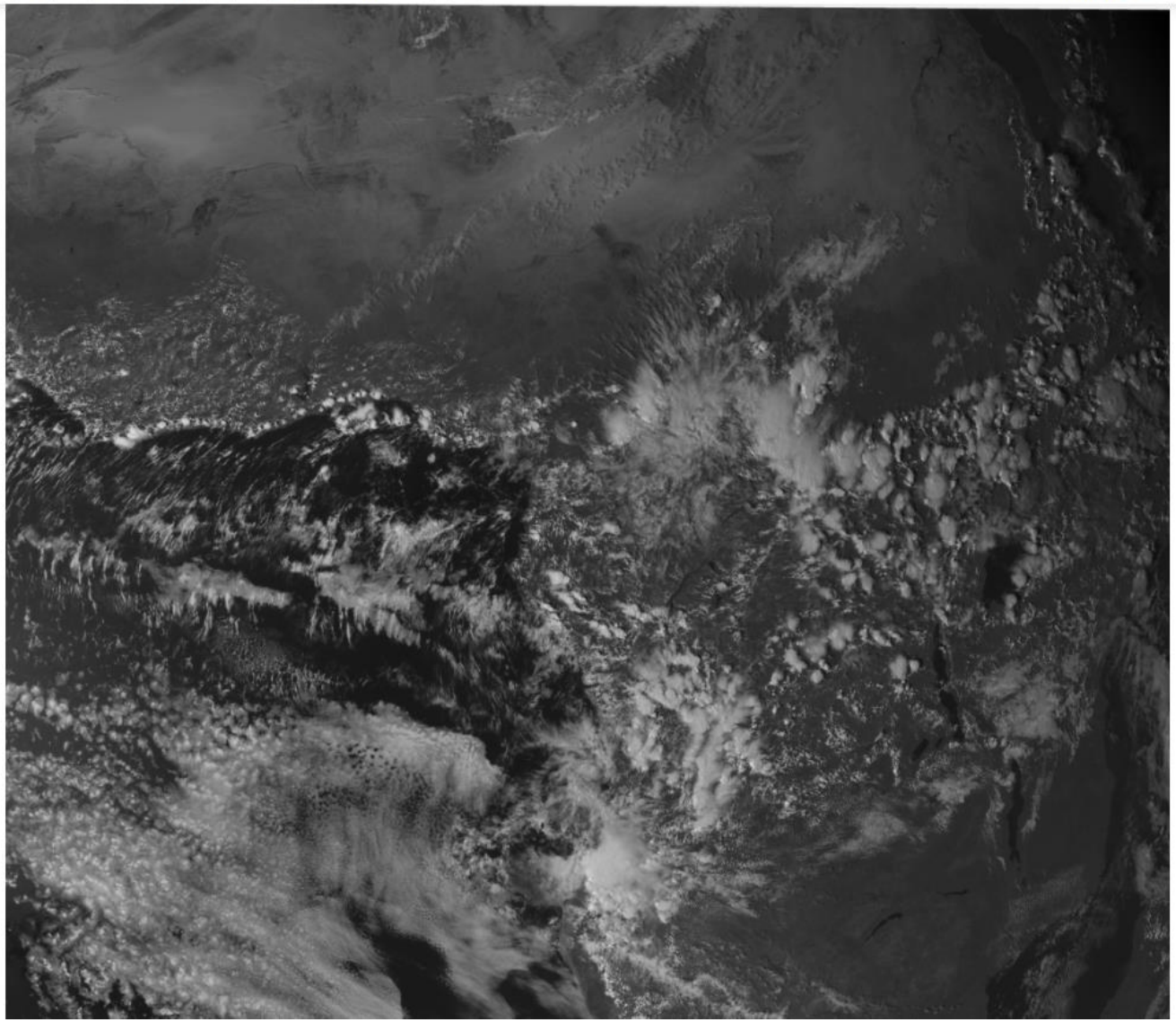
Out[21]:



jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 14 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Markdown Memory: 12.9 GB



jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 14 hours ago (unsaved changes)

Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

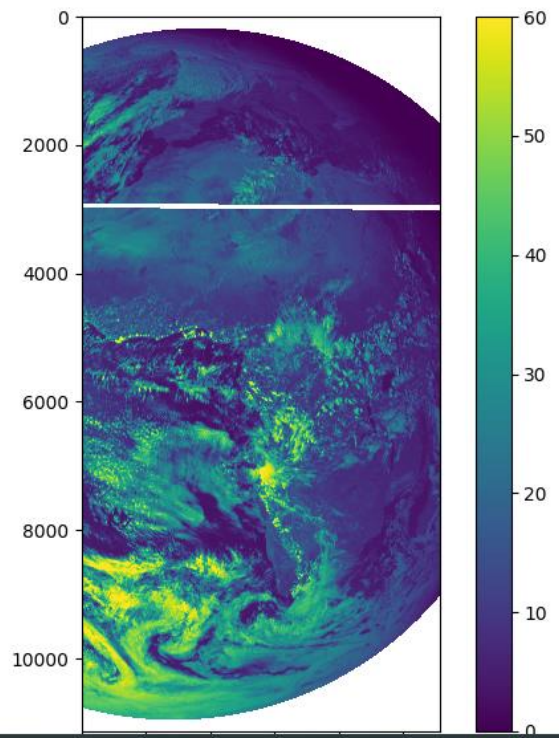
Run Markdown Memory: 13.3 GB

```
When you plotted the figure you coud see that **it is dark, right?**
You can change the value range by adding vmin=0, vmax=60 after ['HRV']
```

In [23]:

```
plt.figure(figsize=(7, 7))

#plt.imshow(scn['HRV'])
plt.imshow(scn['HRV'],vmin=0, vmax=60)

plt.colorbar()
```

Figure 5



jupyterhub Natasa_Data visualisation with SatPy and Jupyter … Last Checkpoint: 14 hours ago Autosave Failed! Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help error Trusted Python 3 (ipykernel)

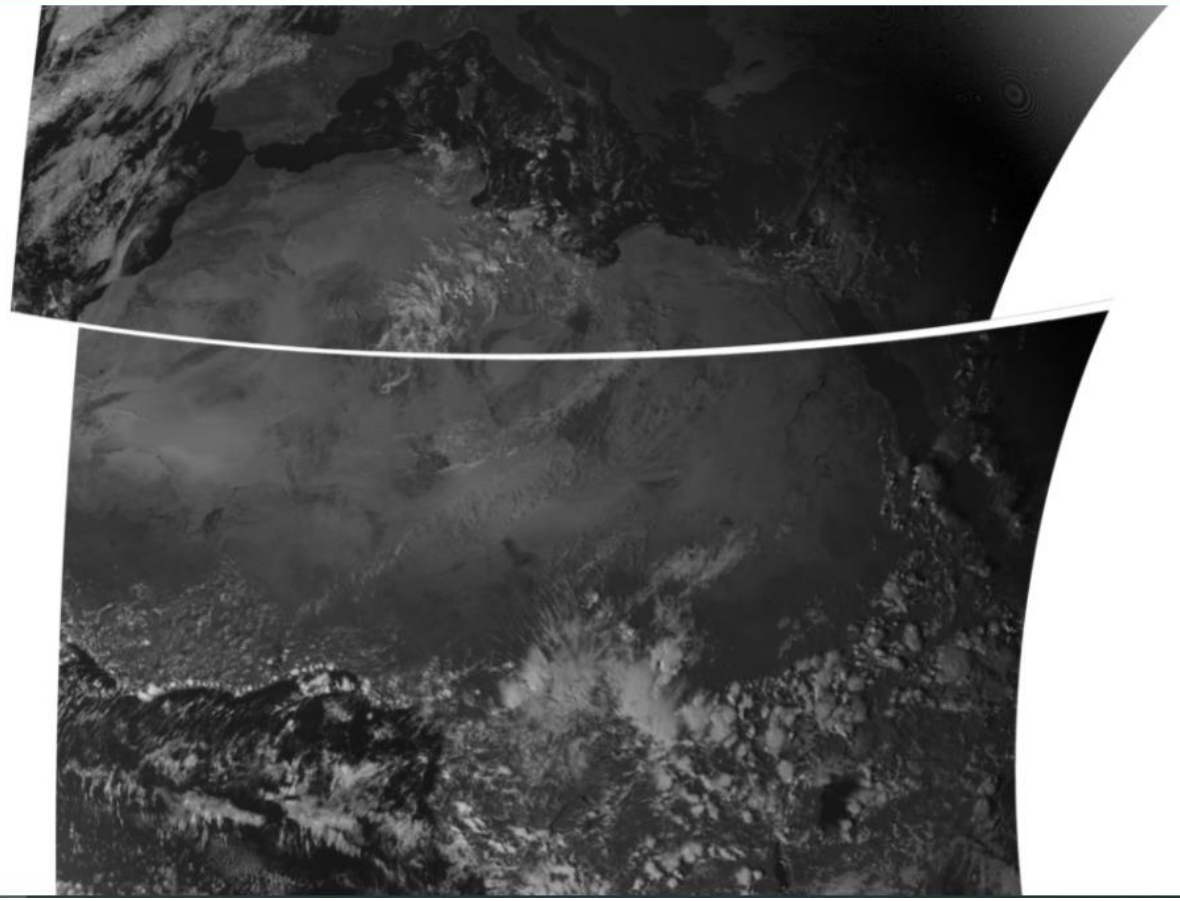
Run Markdown Memory: 13.4 GB

Resample to african continent:

In [24]:
```
1 local_scene = scn.resample("africa")
2 local_scene.show('HRV')
```

```
INFO:satpy.scene:Not reducing data before resampling.
INFO:satpy.resample:Using default KDTree resampler
INFO:satpy.resample:Using default KDTree resampler
```

Out[24]:



jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 14 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help error Trusted Python 3 (ipykernel)

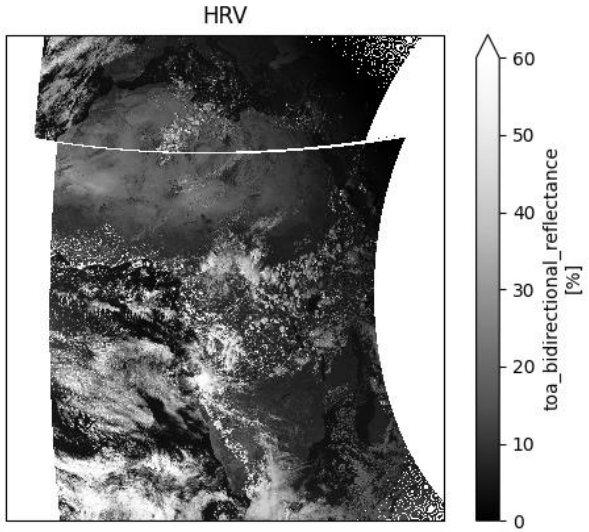
Run Markdown Memory: 14.5 GB

Plot the figure as for IR_108, but with reversed grey scale.

In [26]:

```
1  plt.figure()
2
3
4  crs =local_scene['HRV'].attrs['area'].to_cartopy_crs()
5  ax = plt.axes(projection=crs)
6
7  local_scene['HRV'].plot.imshow(cmap="Greys_r", vmin=0, vmax=60)
8  plt.title("HRV")
9  plt.show()
10
```

Figure 6



jupyterhub Natasa_Data visualisation with SatPy and Jupyter… Last Checkpoint: 14 hours ago (unsaved changes) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help error Trusted Python 3 (ipykernel)

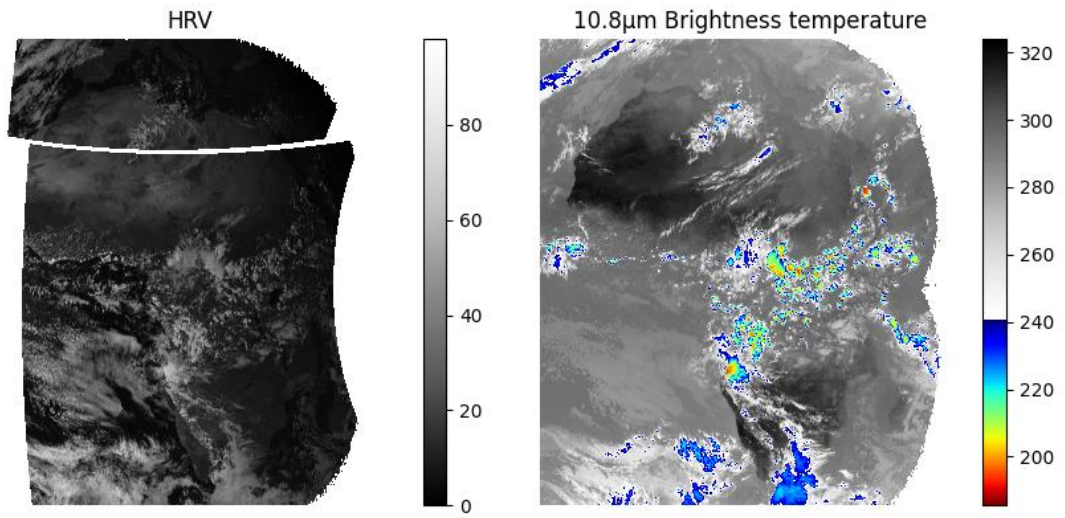
Run Markdown Memory: 16.4 GB

To compare the plots you can use exported plots/figures or plot them side by side (problem with a memory could arise), or plot one below another.

In [29]:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10,6), sharex=True, sharey=True)

im1 = ax1.imshow(local_scene['HRV'],cmap="Greys_r")
ax1.set_axis_off()
im2 = ax2.imshow(local_scene["IR_108"],cmap=newcmp)
ax2.set_axis_off()

ax2.set_title("10.8µm Brightness temperature")
ax1.set_title("HRV")

fig.colorbar(im1, ax=ax1, fraction=.05)
fig.colorbar(im2, ax=ax2, fraction=.05)

plt.show()
```

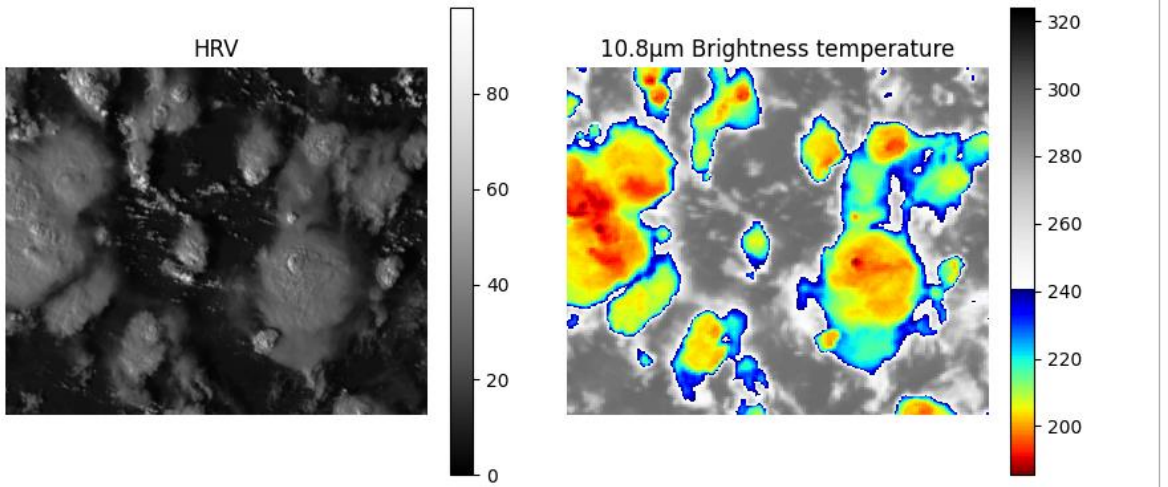
Figure 8



jupyterhub Natasa_Data visualisation with SatPy and Jupyter … Last Checkpoint: 14 hours ago Autosave Failed! Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help error Trusted Python 3 (ipykernel)

Run Code Memory: 16.5 GB



Zoom on the convective storms and compare the HRV and the IR images. What extra features do you see in the HRV image that you cannot see in the IR image

In [ ]:

```
# save the figure. note that we can do it without setting dpi
plt.savefig('HRV vs 10.8.png')
```

Question: conceptually, what feature are the cold storms over northern South Africa and southern Botswana? Can you see any structures? And, what do you think (which synoptic situation) has caused these convective storms?

To answer these questions, you may display the VIS0.8 and WV6.2 images. And, you could display the following composites 'airmass', 'dust' and/or 'hrv_severe_storms'.

Accessing notebooks

- https://trainhub.eumetsat.int/