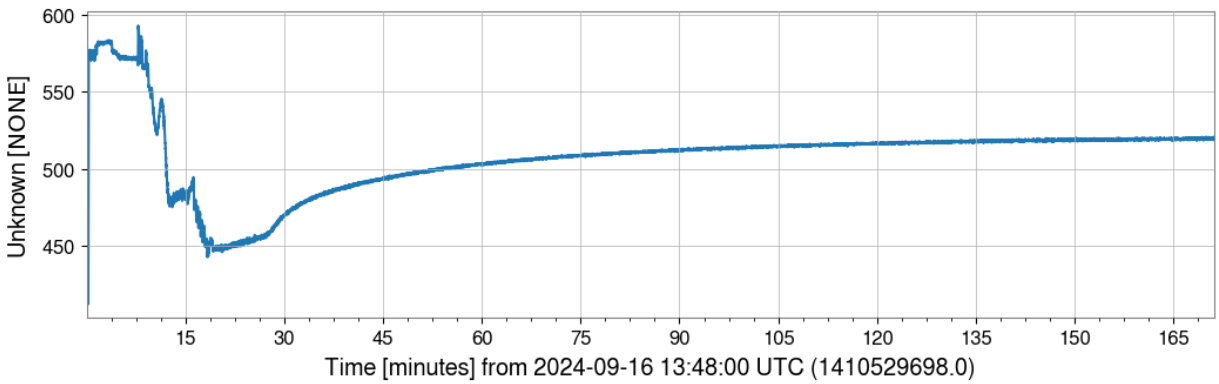
In [1]:
```
from gwpy.timeseries import TimeSeries
from gwpy.plot import Plot
```

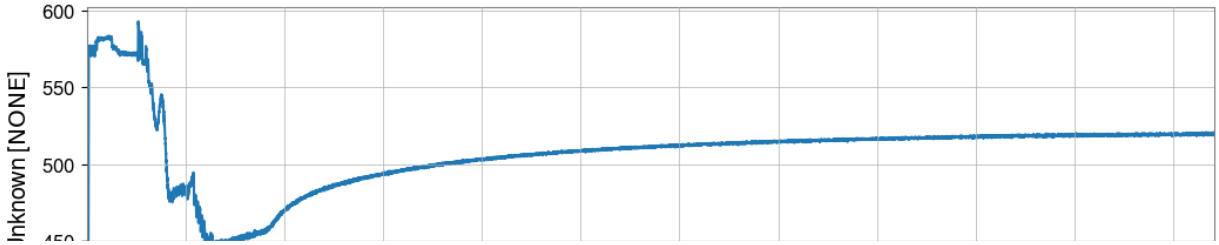
```
/cvmfs/software.igwn.org/conda/envs/igwn-py39/lib/python3.9/site-packages/gwpy
/time/__init__.py:36: UserWarning: Wswiglal-redir-stdio:

SWIGLAL standard output/error redirection is enabled in IPython.
This may lead to performance penalties. To disable locally, use:

with lal.no_swig_redirect_standard_output_error():
    ...

To disable globally, use:

lal.swig_redirect_standard_output_error(False)

Note however that this will likely lead to error messages from
LAL functions being either misdirected or lost when called from
Jupyter notebooks.

To suppress this warning, use:

import warnings
warnings.filterwarnings("ignore", "Wswiglal-redir-stdio")
import lal

  from lal import LIGOTimeGPS
```

In [3]:
```
ReferenceTime = 1410529699
EndTime = 1410539966 # more than 2 hours of thermalization


IFOtrig = 'H1:LSC-IFO_TRIG_INMON'

TrigData    = TimeSeries.fetch(IFOtrig, ReferenceTime, EndTime)
```

In [4]:
```
TrigData.plot()
```

Out[4]:



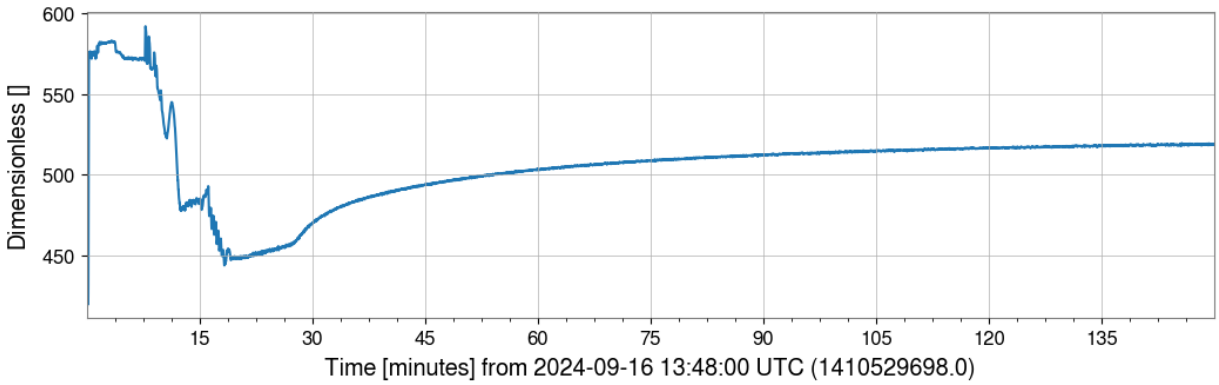


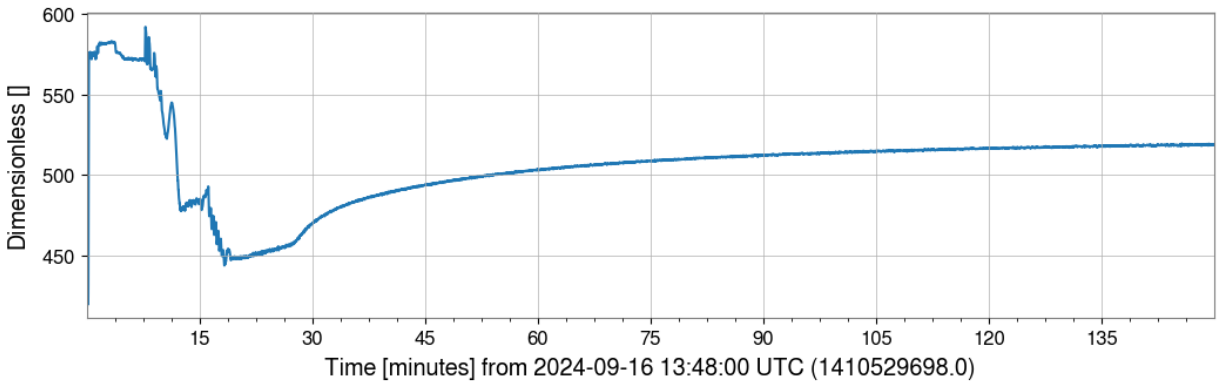
In [29]:

In [43]:

```
window_size = 3 * int(TrigData.sample_rate.value)
Data = TrigData.value[0:150*60*int(TrigData.sample_rate.value)]
i = 0
moving_averages = []
while i < len(Data) - window_size + 1:
    window = Data[i : i + window_size]
    window_avg = round(sum(window) / window_size,2)
    moving_averages.append(window_avg)
    i+=1

avgs_series = TimeSeries(moving_averages, dt = TrigData.dt, t0 = TrigData.t0+1
avgs_series.plot()
```

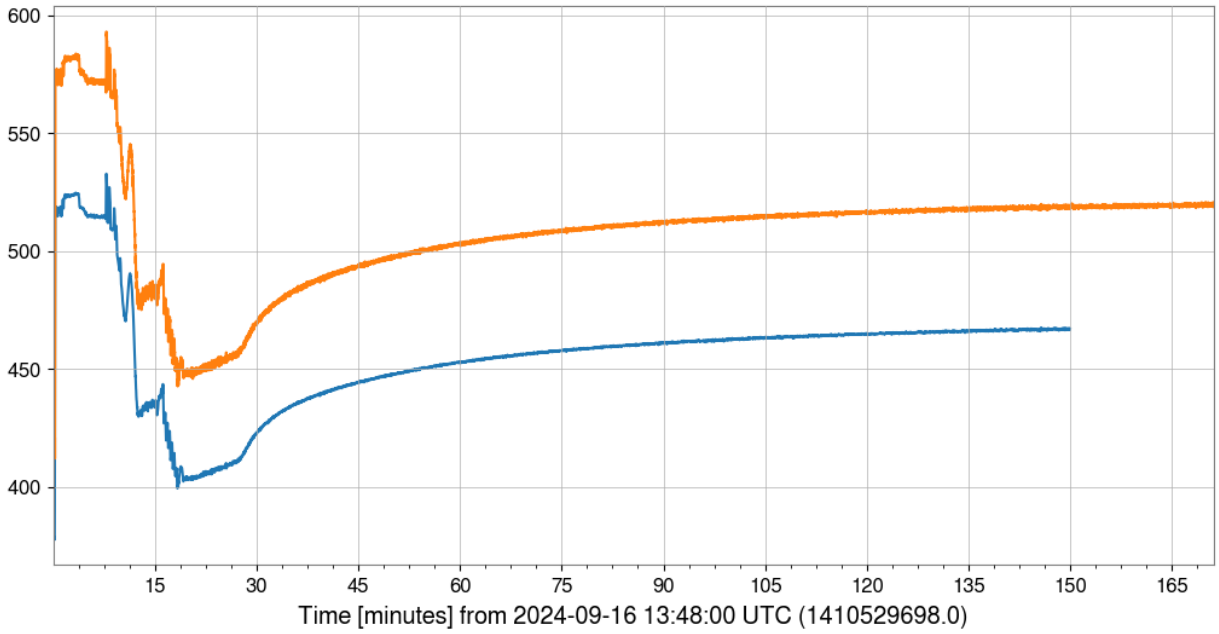
Out[43]:





In [44]:

```
pltobject = Plot(avgs_series*.9, TrigData)
```



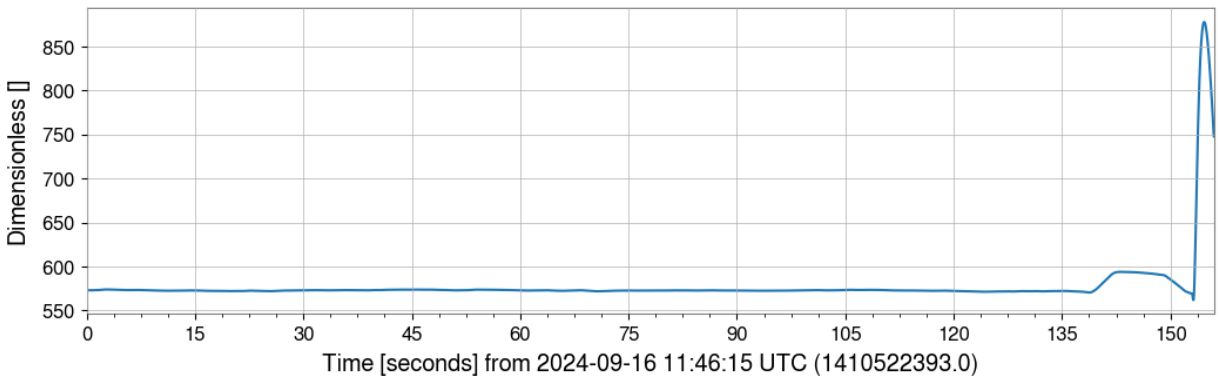
In [45]:

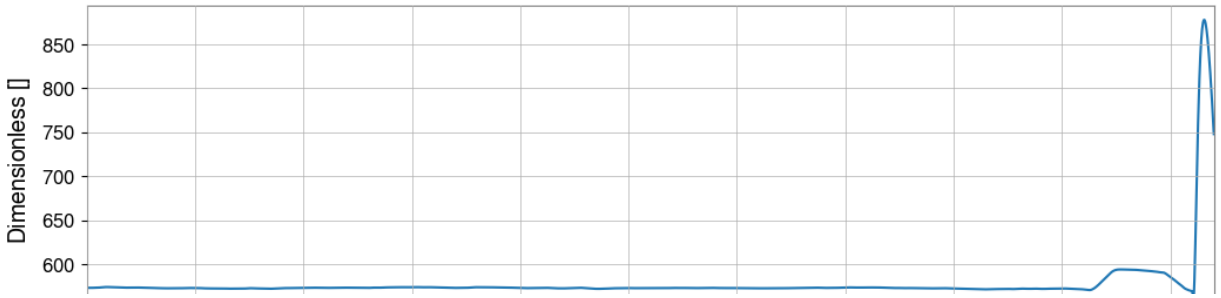
```
ReferenceTime = 1410522390
EndTime = 1410522549


IFOtrig = 'H1:LSC-IFO_TRIG_INMON'

TrigData    = TimeSeries.fetch(IFOtrig, ReferenceTime, EndTime)
```

In [51]:

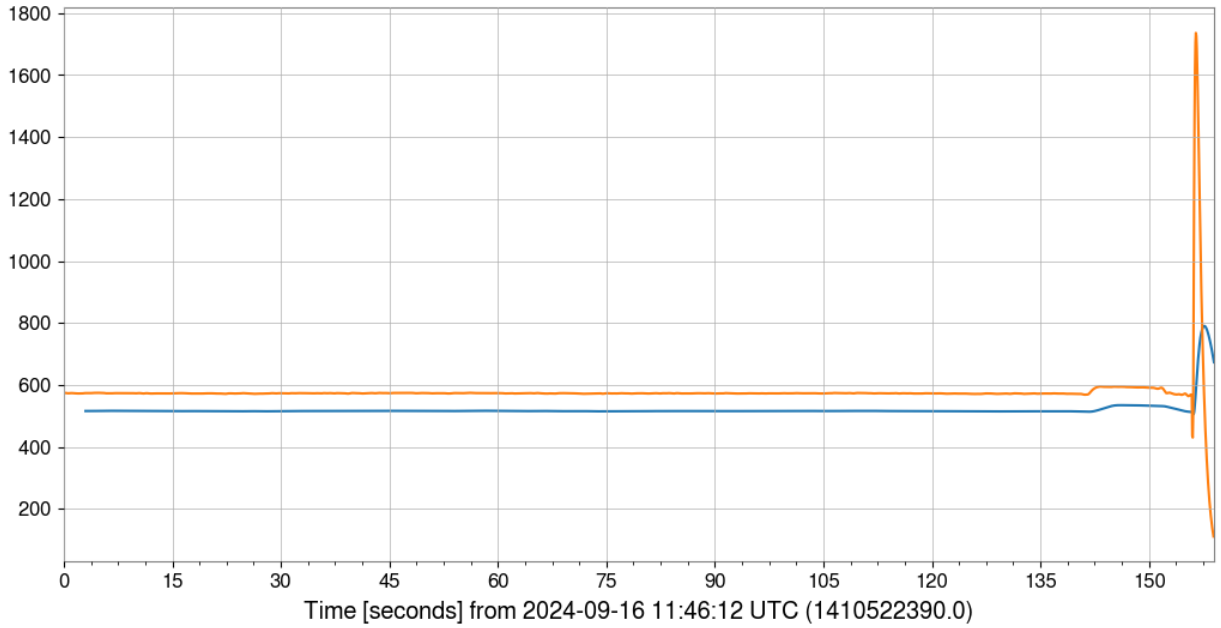
```
window_size = 3 * int(TrigData.sample_rate.value)
Data = TrigData.value#[0:150*60*int(TrigData.sample_rate.value)]
i = 0
moving_averages = []
while i < len(Data) - window_size + 1:
    window = Data[i : i + window_size]
    window_avg = round(sum(window) / window_size,2)
    moving_averages.append(window_avg)
    i+=1

avgs_series = TimeSeries(moving_averages, dt = TrigData.dt, t0 = ReferenceTime
avgs_series.plot()
```

Out[51]:





In [55]:
```
pltobject = Plot(avgs_series*.9, TrigData)
```



In [50]:
```
TrigData.t0
```

Out[50]: $1.4105224 \times 10^{9} \; \mathrm{s}$

In [ ]: