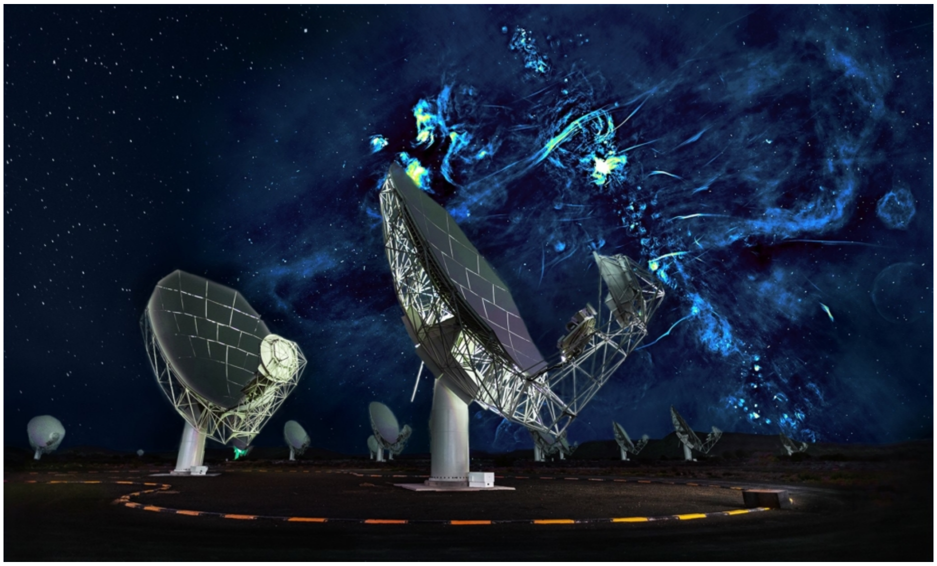
MeerKat Operator Manual

SARAO



Authors: Telescope Operators

DOCUMENT APPROVALS

	Name	Designation	Affiliation	Date	Signature
Prepared By	Ops Team	Operators	SARAO		
Reviewed By	Ops Team	Operators	SARAO		N/A
Accepted By	C Gumede	Telescope Operations	Manager		SARAO
Approved By	L Magnus	Head: Operations	SARAO		

DOCUMENT HISTORY

Revision	Date Of Issue	Prepared By	Comments (e.g. ECN Number or changes to document)
A	12-04-2021	Ops	Issued for Comments & Review

DOCUMENT DISTRIBUTION

Publish in eB and Distribute to all signatories on the document and the relevant line managers.

DOCUMENT SOFTWARE

Package		Version	Filename
Latex	TexStudio	N/A	`https://docs.google.com/document/d/1AOin8UJAxStFA71TCvOO4q_bQ_mOupbCd9ghWZxnRc/edit#`

COMPANY DETAILS

Name	SKA SA, Johannesburg Office (Rosebank, Gauteng)	SKA SA, Cape Town Office (Pinelands, Western Cape)	SARAO, HartRAO (Hartebeeshoek, Gauteng)	SARAO, Karoo Astronomy Reserve (Carnarvon, Northern Cape)
Physical/Postal Address	1^{st} Floor, 17 Baker Street Rosebank, Gauteng 2196, South Africa	3rd Floor, The Park, Park Road, Pinelands, 7405, South Africa	P.O.Box 443, Krugersdorp 1740, South Africa	Posbus 69, Carnarvon, 8925, South Africa
Tel.	+27 11 268 3400	+27 21 506 7300	+27 (12) 301-3100	+27 21 506 7300
Fax.	27 11 442 2454	+27 21 506 7375	+27 (12) 301-3300	+27 (0)86 538 6836
Website	www.ska.ac.za	www.ska.ac.za	www.hartrao.ac.za	www.ska.ac.za

MeerKAT Telescope Operator Manual

Document number	SSA-0006D-003
Revision	A
Classification	Commercial in Confidence
Prepared By	Telescope Operators
Approval Date	01 02 2021

Organisation	:	NRF (National Research Foundation)
Facility	:	SARAO (South African Radio Astronomy Observatory)
Project	:	N/A
Document Type	:	Guidelines (UG)
Function/Discipline	:	Operations Management (0006)

Preface

This manual is designed to help new users (i.e Students, Oservers) of the MeerKat Radio Telescope to understand basic operational procedures, and scientific procedures that are being used to carry out scientific observations. The complexity of the instrument requires many software interfaces for instrument setup, scheduling, monitoring and maintenance management.

The document is an effort to explain the terminologies that are often used locally in operations and commissioning divisions of SARAO. The scientific community is more interested in the scientific methods that were used to calibrate, validate and test all the critical components of the telescope. Thus, important calibration procedures are briefly described with reference to detailed documentation.

Naming Convenctions

The following naming convections have been adopted for this manual:

- Scientific Units [6].
  e.g dB
- Names of servers and computers.
  e.g < *server*>.< *telescope*> . < *site*> *.kat.ac.za.*
  kat@obs.mkat.karoo.kat.ac.za
- Names of software packages.
  e.g `PYTHON`
- Terminal commands and program option. e.g

  ```
  ssh kat@obs.mkat.karoo.kat.ac.za
  ```

- GUI commands e.g
  Select **Links** > **Grafana** > **Overview**
- Folders and Filenames. e.g *`image.py`*
- Names of Meerkat components.
  e.g `m001`
- Names of sensors.
  e.g *ap.control*
- Scripts options and parameters
  e.g `fft-shift`
- Links and urls
  e.g `fft-shift`
- Progress, sensor and system logs

List of Acronyms

ABL	Allocated Baseline
Ac	Critical Availability
ADR	Architecture Design Review
AGN	Active Galactic Nuclei
Ai	Inherent Availability
AOR	Annual Operating Requirement
AR	Acceptance Review
BOM	Bill Of Material
CA	Criticality Analysis
CDR	Critical Design Review
DDR	Detail Design Review
D-Level	Deport Level
DLM	Depot Level Maintenance
FAT	Factory Acceptance Tests
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects and Criticality Analysis
FPGA	Field Programmable Gate Array
FRACAS	Failure Reporting and Corrective Action System
GHz	Giga Hertz
GUI	Graphical User Interface
HartRAO	Hartbeeshoek Radio Astronomy Observatory
Hrs	Hours
I-Level	Intermediate Level
ILM	Intermediate Level Maintenance
ILOR	Intended Learning Outcomes Report
ILS	Integrated Logistic Support
ISO	International Standards Organisation
KAT-7	Karoo Array Telescope, 7 array
Kg	Kilogram
Km	Kilometer
L3/4/5	Level 3/Level 4/Level 5
LEMP	Logistic Engineering Management Plan
LRU	Line Replaceable Unit
LSA	Logistic Support Analysis
MBL	Manufacturing Baseline

MSCDR	Media Selection & Curriculum Development Report
MSP	Maintenance & Support Plan
MTBCF	Mean Time Between Critical Failures
MTBF	Mean Time Between Failures
MTTRc	Mean Time To Repair Critical
MTTRi	Mean Time To Repair Inherent
NQF	National Qualification Framework
OEM	Original Equipment Manufacturer
O-Level	Organisational Level
OLM	Organisational Level Maintenance
OTLR	Operator Task List Report
PBL	Product Baseline
PBS	Physical Breakdown Structure
PC	Printed Circuit
PDR	Preliminary Design Review
PHS and T	Packaging, Handling, Storage and Transportation
PPPM	Preparation, Preservation, Packaging & Marking
PPPR	Personnel Performance Profile Report
PRR	Production Readiness Review
PSS	Product Supplier Support
QBL	Qualification Baseline
RAM	Reliability, Availability, Maintainability
RBL	Requirements Baseline
Relc	Reliability Critical
Reli	Reliability Inherent
RF	Radio Frequency
RFI	Radio Frequency Interference
RM	Rotation Measures
RR	Requirements Review
RTS	Receptor Test System
S and TE	Support and Test Equipment
SAQA	South African Qualifications Authority
SEMP	System Engineering Management Plan
SKA	Square Kilometer Array
S-Level	Supplier Level
SLM	Supplier Level Maintenance
SNR	Supernova Remnants
SRU	Shop Replaceable Unit
TBD	To Be Determined
TRR	Test Readiness Review
TSR	Training Survey Report
TTLR	Technical Task List Report
vs	Versus

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The MeerKAT telescope consists of 64 receptors with 13.5 m diameter dishes, designed to achieve high sensitivity and imaging dynamic range, while providing an array and functionality to provide for a wide range of science. MeerKAT achieves a sensitivity of at least 220 m2/K at L-band (our designs indicate that we may achieve 300 m2/K in the L-band).
The high sensitivity requirement drives the design to multiple octave band single pixel feeds with cryogenic cooling. Three receivers cover the required operating band in the frequency ranges 0.58 - 1.015 GHz, 0.9 - 1.67 GHz and 8 - 14.5 GHz. The offset Gregorian dish configuration enhances sensitivity by providing high aperture efficiency and low spill-over temperature contribution. The offset Gregorian dish configuration also provides a clean optical path that can be designed to produce low overall sidelobe levels and azimuthal symmetry in the inner sidelobes to achieve high imaging dynamic range.
Low sidelobe levels also provide good rejection of unwanted radio frequency interference from satellites and terrestrial transmitters. The array configuration is designed with two components: a compact core containing 70 % of the dishes, and an extended array designed for high fidelity imaging performance over a range of resolutions from 6 arcsec to approximately 100 arcsec.
The overview of the MeerKAT telescope system illustrating major components and their functions are shown in **Figure** 1.1.

1.1 Antenna Positioner

The MeerKAT telescope array consists of 64 Receptors, each consisting of a steerable dish antenna, a set of radio receivers and a set of associated digitisers. The digitised radio signals are then transported over an array fibre network back to the central on-site data centre at the Karoo Array Processor Building (KAPB) where they undergo various stages of processing (correlation, beam forming and science processing) as shown in **Figure** 1.2. The telescope array will be controlled and monitored from a number of remote locations, with the main control operations centre in Cape Town. The Receptor will be constituted by a Digitiser, an Antenna Positioner and a Receiver System.

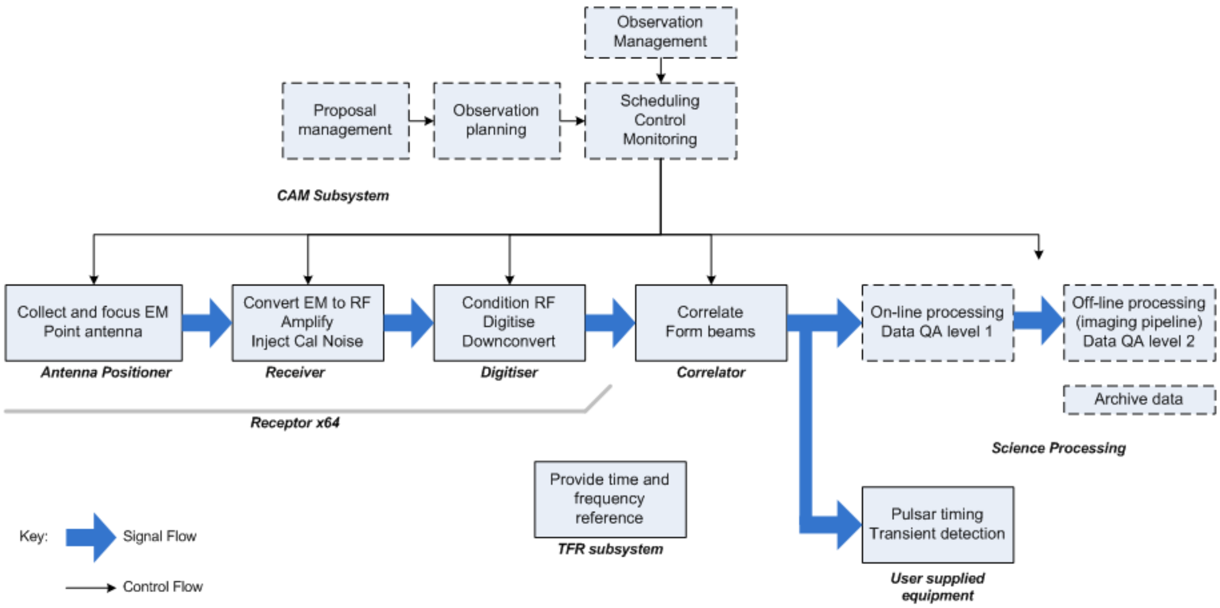


Figure 1.1: Overview of MeerKAT System

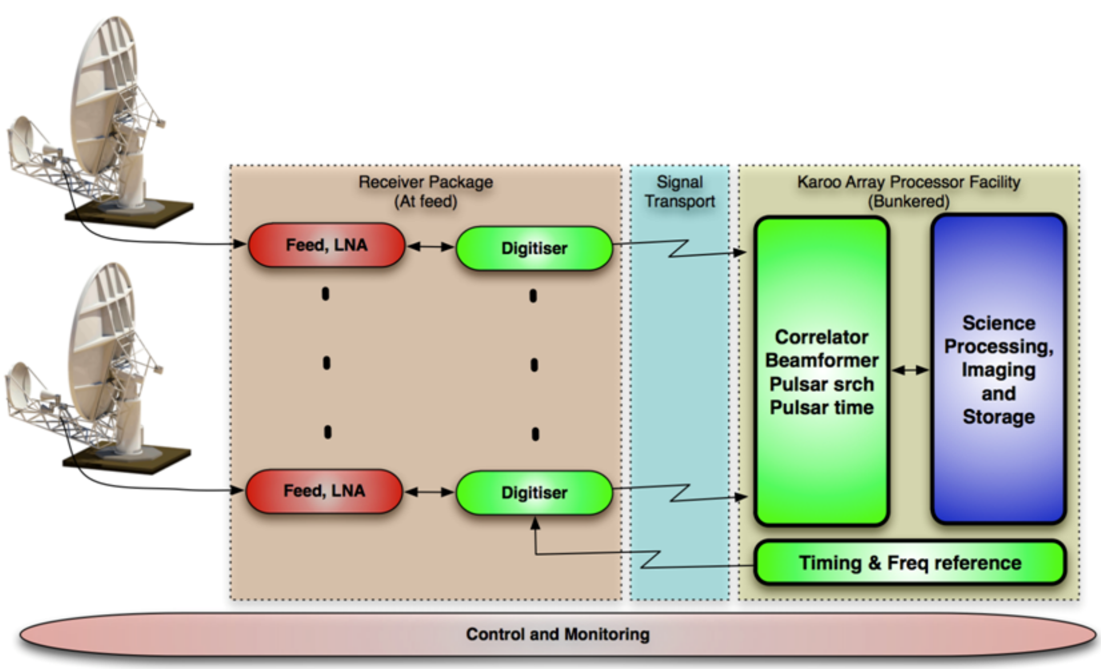


Figure 1.2: MeerKAT telescope signal path

1.2 Antenna Positioner

The Antenna Positioner (AP)[2] main function is to intercept flux. This function will be provided by the Antenna Structure, Pointing Control System and Receiver Indexer major components of the AP. An offset Gregorian optical layout with a 13.5 m projected diameter aperture is chosen as it offers several advantages. The unblocked aperture of the offset design provides uncompromised optical performance and sensitivity, excellent imaging quality, and good rejection of unwanted radio frequency interference from satellites and terrestrial transmitters. This layout also provides space for a suite of receivers mounted on an indexer. The Antenna Structure will be constituted by a dual reflector system mounted on an Elevation over Azimuth mount. The offset reflector will be part of the Reflector mounted on a Yoke. The Yoke provides the elevation and azimuth rotation to achieve pointing over the required observation range. Azimuth rotation is achieved by means of an Azimuth Drive and a geared Azimuth Bearing between the Yoke and the Pedestal. Pointing coordinates in the form of azimuth- (referenced from true North) and elevation values from the Telescope Control and Monitor subsystem (CAM) are received and interpreted by the Position Controller, and executed by the Elevation and Azimuth Drives of the Pointing Control System. The Antenna Positioner provides monitoring information to the Telescope and implements a standardised monitor and control interface (KATCP) over TCP/IP. The 4 receivers of the Receiver System are mounted on the Receiver Indexer of the Antenna Positioner[TBD].

1.3 Receiver System

The MeerKAT receiver system[7] consists of the following major components: a UHF-band receiver, an L-band receiver, a vacuum pump station, a helium compressor and a receiver system controller (RSC). The receivers and vacuum pump station are mounted onto the receiver indexer, the helium compressor is mounted onto the yoke and the RSC slots into the 19” rack in the shielded drive compartment of the antenna pedestal. The receiver system is designed to operate with 4 receivers and thus makes provision for 2 more receivers to be added in the future. The receivers use Gifford-McMahon (GM) cryogenic cooling which significantly exceeds the current sensitivity specification[TBD].

1.4 Digitiser System

The main function of the Digitiser[5] is to digitise the analogue RF signal provided by the receivers. RF signals from both polarisations of each feed enter the appropriate Digitiser and get conditioned via an analogue RF conditioning section before entering the ADC. The RF conditioning section filters and amplifies the signal before entering the ADC. Filtering is performed in such a manner that Digitiser efficiency as well as out-of band suppression requirements are achieved. Further functionality of the RF conditioning section is to provide adjustable gain to improve telescope operational dynamic range, power detection of incoming and pre-ADC RF levels and potential bi-phase modulation to improve RF channel isolation. All analogue electronics preceding the ADC is potentially thermally controlled to the temperature that results in the best phase and amplitude stability of the RF signal. An ultra high speed ADC samples the conditioned RF based on a sample clock that is provided by the TFR subsystem. High speed optical interfaces between the ADC and the D engine ensure that RFI generated by these assemblies do not degrade the Digitiser RFI performance. The

D engine receives the digitised data from the ADC and depending on what Nyquist zone is being sampled, down-convert the entire band to baseband frequencies using a DDC with programmable LO. The down-converted signal is packetized and transported to the Correlator data switch via Ethernet fibre optic links. All data products are time stamped with the Digitiser local time which is derived from the 1 pulse per second (1PPS) signal provided by the TFR via analogue fibre. Control and Monitoring of the Digitiser as well as data transport from the Digitiser to the Correlator is performed on the same optical Ethernet interface [TBD].

1.5 CAM System

The CAM subsystem[9] is responsible for Control and Monitoring of the MeerKAT telescope and presentation of the CAM user interfaces for operators, engineers and science users. The CAM subsystem shall monitor all monitoring points for health, state and alarms, archive history of all monitoring points and allow easy interrogation of the monitoring archive. The CAM subsystem shall provide a Proposal Management Tool that will support the proposal process from application submission through to final approval, as well as provide mechanisms required by the Observation Planning Tool (responsibility of the Science Processing Team) [1]. The Observation Planning Tool (OPT) will support the observer in preparing the program and scheduling blocks for an approved proposal and may need mechanisms from CAM e.g. to simulate or dry-run observation execution. The CAM subsystem shall control the telescope during observations including scheduling, configuring and controlling other subsystems, monitoring execution of programs, noting the data products produced and preparing an observation report on completion of each program.

1.6 Correlator BeamFormer

The Correlator-Beamformer (CBF)[8] is configurable for a range of data products to support all the required types of observations. This includes spectral line and continuum imaging, pulsar timing, tied-array beamforming and various transient search functions. It uses a packetized architecture comprising switches and reconfigurable modular processing elements, which makes it flexible, upgradeable and scalable.

1.7 Science Processor

The Science Processor (SP)[1] is a subsystem of the MeerKAT telescope whose primary role is the production of science quality output from the raw data provided by the Correlator-Beamformer subsystem. The SP includes both visibility and time-domain processing capabilities, along with an extensive archiving component, and a variety of user interfaces, including the primary point of interaction with the science users in the form of the Observation Planning Tool. The SP can be regarded as the terminal system of the telescope, with downstream interactions leaving the boundaries of the telescope system itself.

1.8 Time and Frequency Reference System

The Time and Frequency Reference (TFR)[3] is an essential subsystem of the MeerKAT, in providing absolute traceable time to the system using atomic clocks. This is necessary in nearly all the telescope functions. Frequency directly impacts the quality of data and time itself forms part of the telescopes signal data products, as this measurement together with other data, enables specific science including precision astrometry. This specific requirement specification document largely focuses on a correct implementation for the L-band, but equipment specification for equipment clocks is done, so as not to limit adaptability to other bands including S-band and X-band. Furthermore the pulse per second specifications is applicable to any band. Certain specifications are however left open for the S and L-bands. The function of the TFR subsystem is to provide and distribute accurate time to the Telescope system. It is also expected to provide both PPS and PTP to be used by other subsystems of the telescope.

Chapter 2

Logging into the system

The telescope system software runs on Linux servers and the basic understanding of the Linux operating system is required in order to operate the telescope. The Telescope Operator will also use Portal's CAM graphical user interface referred to as GUI in order to interact with the telescope system. In order for anyone to interact with the telescope system, they need to have been given access via CAM portal. The person needs a username and password to login into the system. When a person logs in for the first time on the system in order to control and monitor MeerKAT telescope system, the following is important to note. The system has many server nodes that are used for controlling and monitoring various parts of the system. The most used servers nodes are:

- *portal.mkat.karoo.kat.ac.za* and
- *obs.mkat.karoo.kat.ac.za*

The naming convention for the server nodes is always
< *server*>.< *telescope*> . < *site*> *.kat.ac.za.*, with :

- the first parameter is server node name
- the second is the telescope (Currently: *mkat, kat7, mkat-rts*) and
- the third is karoo for site systems different from lab systems.

2.0.1 Portal Server

The URL of where to find the GUI is *http://portal.mkat.karoo.kat.ac.za/*. The login details to the portal link above will be provided by CAM admistrator. The landing page is as shown in **Figure** 2.1.

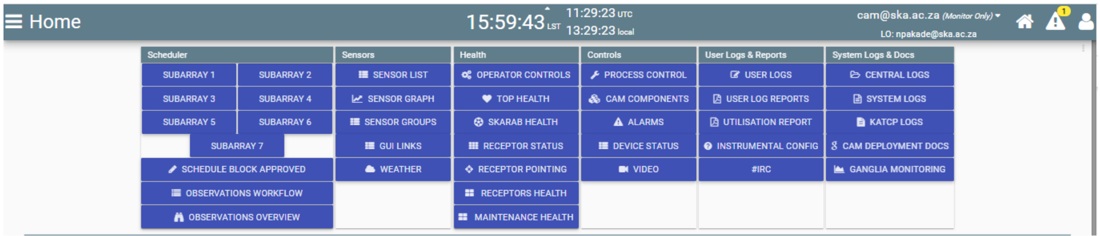


Figure 2.1: CAM GUI home page

This will allow one to monitor the telescope system but will not be able to control the system. The person will have to be given the lead operator role (LO) or expert role in order to control the telescope for safety reasons. This page will allow one to access different functionality of the telescope system, monitoring and control i.e. if for instance one is interested in building and running subarray 1, the configuring of a subarray, one will click and open "SUBARRAY1 tab. This page will be shown as in **Figure** 2.2 below.

Figure 2.2: CAM GUI Subarray1 window

In order to learn more about the GUI and the telescope system, the user must familiarise with different tabs on the GUI. More information will be discussed in the following sections of this document.

2.0.2 Obs Server/Machine

The node server *"obs.mkat.karoo.kat.ac.za"* allows interaction with the live system via the command line and ipython interface. This is where the instructions to create a schedule block SB is provided.The user is required to command the telescope in the Linux environment from the command line using Linux and python commands.

The user will be given credentials to login into the server. The server node to login using ssh command is *"obs.mkat.karoo.kat.ac.za"* and this will open up a page as shown in **Figure** 2.3.

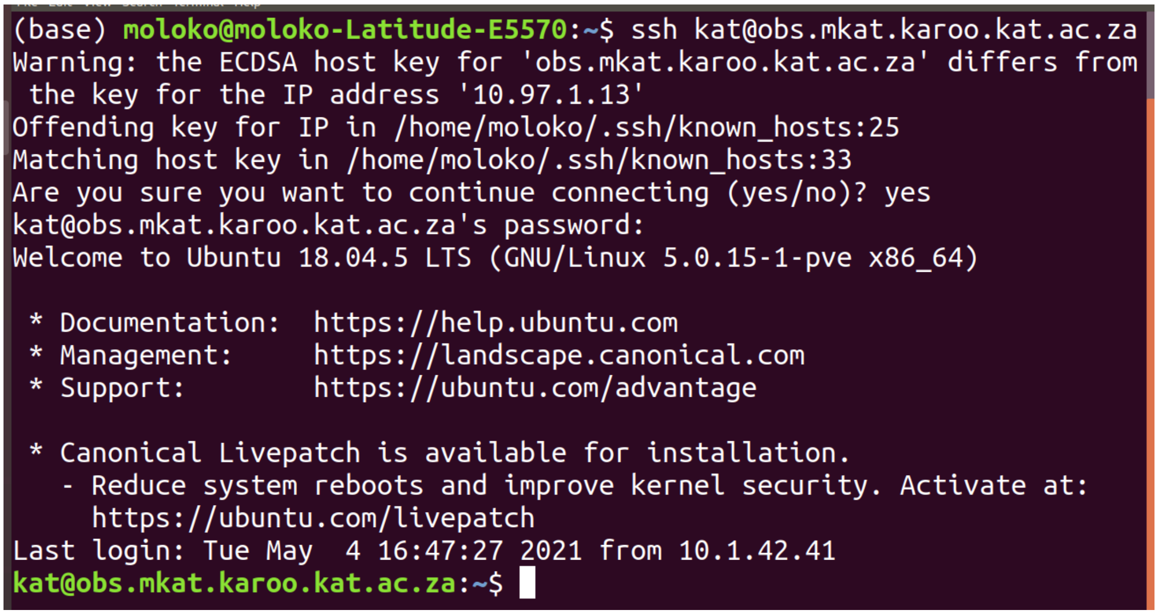


Figure 2.3: Terminal ssh login to obs machine

Chapter 3

Handover Procedures

The telescope can only be controlled by a lead operator who has the correct credentials. To be a lead operator you will have credentials to control and monitor the telescope and these credentials are supplied by the Telescope Operations Manager. The telescope operates 24 hours a day, 7 days week and 365 days a year and there are 3 shifts a day split over 8 hours each. During the shift we have a lead operator who is responsible for controlling and monitoring the telescope system. Therefore there are three operators over a 24 hour period configuring and monitoring the telescope system at all times. Between the shifts one lead operator hands over the system to the next operator. The paragraphs below describe in detail the procedure begging followed during the handover process.

3.1 Operator Handover Procedures

Each Operator during the shift compiles a list or record of the incidents that occured during shift. This record is called a handover document and this is different from the operations meeting minutes. The operator will update the incoming lead operator and also give this document to the next lead operator and this document will contain but not limited to this information below:

- What is currently being done with the telescope?
- What observation is coming up after the current one?
- What receptors are available and what receptors are expected back from maintenance?
- What resources have been booked out for. maintenance and when - who to contact to get them back?
- What problems are currently opened under which subsystems - supplement? this with reading open-ended user logs and the checklist [**Appendix A**]"
- What are the problems experienced and what does the incoming operator need to look out for?

3.2 Booking resource for maintenance

The lead operator during the morning shift will be required to make resources (AP, Receivers, Digitiser, CBF etc) available to site technicians and engineers to conduct maintenance and upgrades. The resources will normally be booked for maintenance or upgrade at least a day before unless the resource is in error and cannot be used for operations. The lead operator will follow the process below:

- Check for booking in the Engineering Spreadsheet or Site calendar.
- Mark the resource to maintenance - a log is automatically opened in the user logs. Add reason for booking in the log.
- If the resource booked causes system downtime, add the 'timeloss' tag to the user log for system utilisation statistics reporting.
- Alert maintenance staff or engineering staff after the resources have been made available.
- This information must also be added in the handover document.

3.3 Receiving a resource after maintenance or an update

- During the handover, check the control status of the resource *ap.control* sensor in sensor list as shown in **Figure** 3.1.
  - If it is in remote - usable
  - If it is in local/manual/e-stop - must be reset by technician

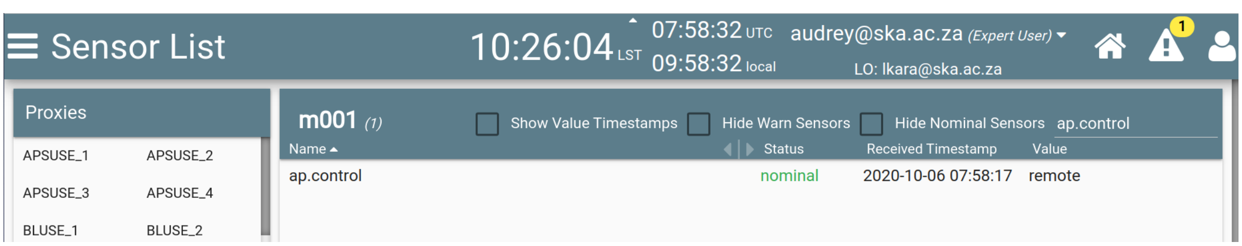


Figure 3.1: CAM GUI sensor list filtered with "Control"

- If in remote, check the health of the resource in detail

```
ssh kat@obs.mkat.karoo.kat.ac.za
./katsdpscripts/utility/check_ant.py --observer name --ant m0xx --proposal-id 22
```

  Or

```
./usersnfs/tiyani/meerkat_status.py --receiver rxl
```

(For L-band receivers)

```
./usersnfs/tiyani/meerkat_status.py --receiver rxu
```

(For UHF receivers)

- Include the AP in the next observation to check signal quality
- If a number of antennas are returning from maintenance and there is time before the next observation, create a small subarray with at least 4 antennas and run delay calibration, if there are less than 4 antennas available, build a subarray with those antennas but run three calibration script to check signal health.
- Update OPS Catalyst page with correct number of APs available for use.

3.4 Integrating a Receptor into Meerkat

This must be done when an antenna has been in maintenance for an extended period of time or it was handed over to MPI for testing

Verify with the person who took the AP if they have changed anything in the digitiser,receiver or any other component configuration. If there were changes made, talk to CAM to change what has been changed before testing the AP. Run the *"check_ant.py"* script as in the step above in 6.2 Mark the digitiser ready

```
ssh kat@obs.mkat.karoo.kat.ac.za
ipython
import katuilib
configure_cam("camcam","all")
cam.m0xx.sensor.dig`_selected_band.get_value()
cam.m0xx.req.digitiser_ready('l', timeout=60)
cam.m0xx.req.dig_select_band('l', timeout=60)
```

Where `m0xx` represents a receptor number e.g `m001`

- Sync the digitiser using global sync - see next chapter.
- Check the remaining hours left before the next global sync in sensor list of the CAM GUI as shown in **Figure** 3.2.

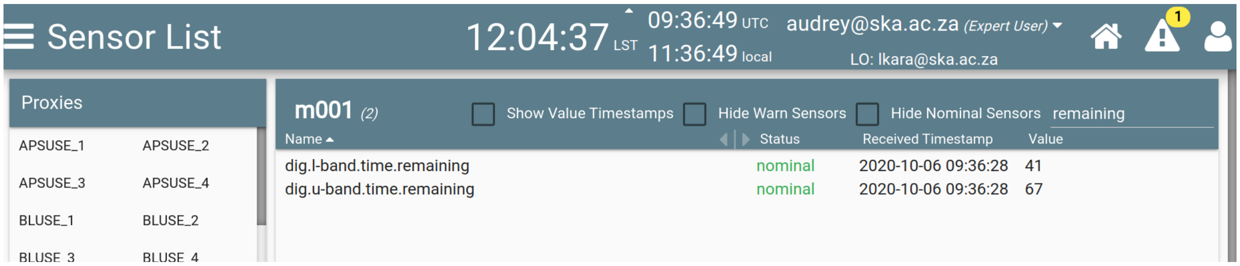


Figure 3.2: CAM GUI sensor list filtered by "remaining".

These values above (41 & 67) will indicate if global sync was successful or not for L and UHF−band digitisers.

- Include the antenna in an array and run a calibrated delay script and see if it finds a solution.
  - Look at the progress output of the delay cal script if the delay solutions are found.
  - During observation, look at the waterfall plot(signal displays) to see if there is a constant colour on its baseline. If the antenna looks noisy as in **Figure** 3.3 or has a rainbow, ask AoD to check its delay models.

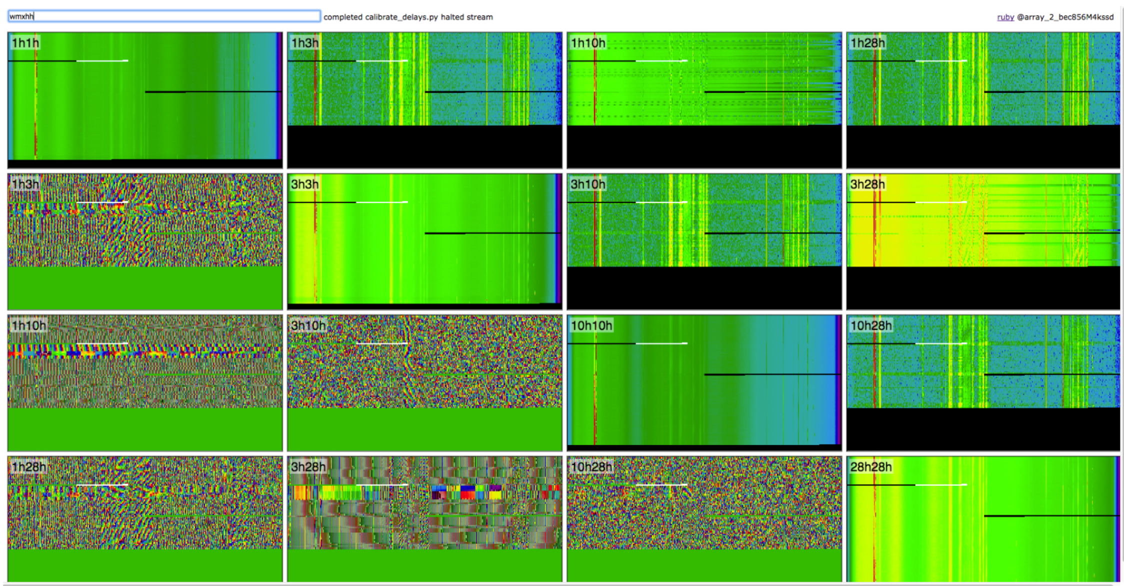


Figure 3.3: SDP waterfall plot.

- Before including the antenna in a Science observation communicate with the AoD about:
  - interferometric pointing and notify operators to update pointing and delay models.
- Update the Engineering Meerkat spreadsheet in the Owner tab.
- Update the associated jira if necessary and close it.

Chapter 4

System Settings Procedures

4.1 Moving Skarabs

WARNING: *do not try to move the skarabs right after stopping or starting a subarray. The SKARABs need a couple of minutes to restart. Otherwise, they will not be found by the script, and will be left behind on the unwanted cmc.*

4.1.1 CMC IP addresses

The IP addresses for different machines are:

cmc1: 10.103.254.1 and name: *cmc1.cbf.mkat.karoo.kat.ac.za*

cmc2: 10.103.254.3 and name: *cmc2.cbf.mkat.karoo.kat.ac.za*,

You can use the IP addresses or hostnames interchangeably (whichever you prefer)

The cbf_support repository is on git. You can clone it from:
git clone `https://github.com/ska-sa/cbf_support`

4.1.2 Check the number of SKARABS

In order to check the number if SKARABS are available in the CMC use the following commands on the obs machine before attempting to move them. There should be about 220(as at 2020-01-06, but more will come) if they were all moved. If there are less, wait a couple of minutes for them to restart.

```
kcpcmd -t 10 -s cmc1.cbf.mkat.karoo.kat.ac.za:7147 resource-list | grep "up$" | wc -l
```

(this will give you the number of skarabs available on cmc1.)

```
kcpcmd -t 10 -s cmc2.cbf.mkat.karoo.kat.ac.za:7147 resource-list | grep "up$" | wc -l
```

(this will give you the number of skarabs available on cmc2.)

4.1.3 Moving SKARABS

If moving to cmc1 use -m cmc1:

```
./usersnfs/cbf_support/./cmc_manage_skarabs.py -m cmc1 -a 5 6 7 8 9 10 11 12 13 14 15 16 17 18
 -k cmc1 cmc2
```

If moving to cmc2 use -m cmc2:

```
./usersnfs/cbf_support/./cmc_manage_skarabs.py -m cmc2 -a 5 6 7 8 9 10 11 12 13 14 15 16 17 18
-k cmc1 cmc2
```

This will connect to all the switches, discover which skarabs are currently online on the various ports, and move them to the requested master controller (-m switch).

4.2 Global Synchronisation

This script seeks to synchronize all digitisers to the Digitiser Master Controller so that signal/data coming into the correlator is in sync and correlates.

- Ensure epoch sync on all usable digitisers (all bands) is done for the day
- In the GUI, verify that all subarrays are inactive:

  ```
  ssh kat@obs.mkat.karoo.kat.ac.za
  run /home/kat/katsdpscripts/utility/global_sync.py
  --observer= name --proposal-id=OPS-23
  ```

- To check that all digitisers are synced and have the same epoch time, in the same ipython session

  ```
  ipython
  import katuilib
  configure()
  kat.report_sensors('epoch','all')
  ```

- Three ways to check time remaining for the next global sync
  - Check the Next Global Sync from the active subarray in the GUI (see **Figure** 4.1). This gives indication on what is the time remaining before the next global sync is done.

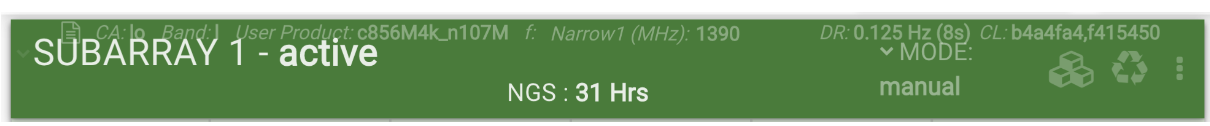


Figure 4.1: Global sync time remaining alert on the GUI

  - From the sensor list by typing *remaining* as shown in **Figure** 4.2.

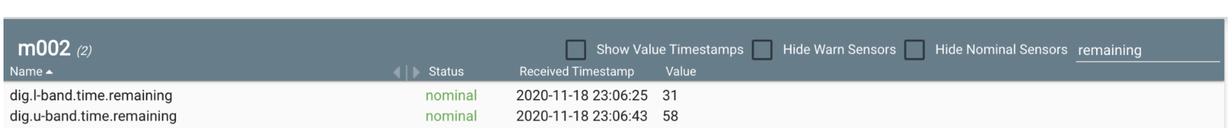


Figure 4.2: Global sync time remaining alert in sensor list

- On IRC there is a reminder that reminds when to run a Global sync counting down hourly from 15 hours left .

4.3 Setting up digitisers

4.3.1 Configured Ipython Session

```
ssh kat@obs.mkat.karoo.kat.ac.za
ipython
import katuilib
configure_cam('camcam', 'all')
```

4.3.2 Mark Digitiser Absent

The digitiser will be marked absent if there is maintenance work on the AP which may cause the power to be switched off to the digitiser. It is also recommended that it is marked absent if the AP will be out for maintenance for a long time. In a configured ipython session run the following commands:

```
cam.m0xx.req.dig_select_band('0')
cam.m0xx.req.digitiser_absent('l', timeout=60)
```

In this case ‘l’ represents L-band digitiser

4.3.3 Mark digitiser ready

When the digitiser was marked absent or the new digitiser was installed in the AP, it is required that it must be set ready for operations. Still in a configured ipython session run the following commands:

```
cam.m0xx.sensor.dig_selected_band.get_value()
cam.m0xx.req.digitiser_ready('l', timeout=60)
cam.m0xx.req.dig_select_band('l', timeout=60)
```

If all digitisers were marked absent previously, this command above must be run for all different bands, i.e. ’l’ must be replaced with ‘u’ for UHF band digitiser. There is no need to set an S-band digitiser at this stage.

4.4 Requesting which receptors have UHF−band digitisers

In order to determine which antennas have UHF band digitisers installed on them, you will run the following commands: On the machine: ssh kat@obs.mkat.karoo.kat.ac.za and run the following commands.

```
u=`kcpcmd -t 60 -s 10.103.254.2:7147 list-digitisers | grep "u as
ready" | cut -d ' ' -f 2` && echo $u
```

If you prefer a list format do the following:

```
u=`kcpcmd -t 60 -s 10.103.254.2:7147 list-digitisers | grep "u as
ready" | cut -d ' ' -f 2` && echo -e "\n$u\n"
```

4.5 Digitiser Health

In order to see the status of each digitiser that is operational one can use the following commands to check for errors and health of each digitiser. For L-band:

```
kat@obs.mkat.karoo.kat.ac.za:~$ dig-stats
```

For U-band:

```
kat@obs.mkat.karoo.kat.ac.za:~$ dig-stats 10.103.254.2:7147 u
```

4.6 Power sensors

Important digitiser sensors to note for operations are power level The digitiser receives RF signals from the receiver and the levels are measured at the inputs of the digitiser by the RFCU. From the CAM GUI sensor list, filter for rfcu i.e. rfcu.*.power. **Figure** 4.3 shows the power levels for each polarisation of m006.

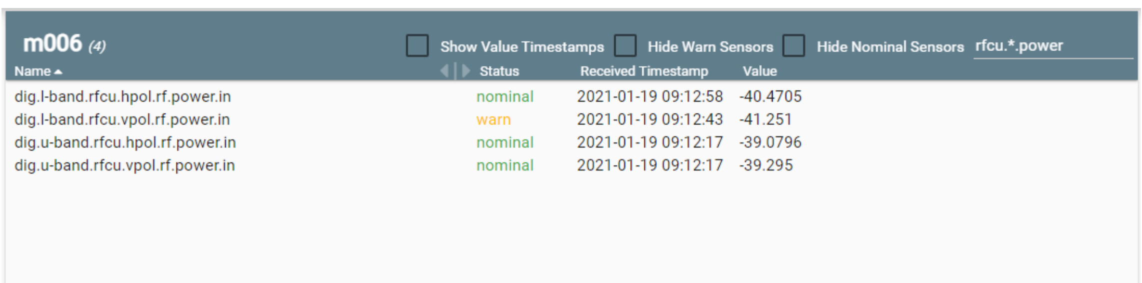


Figure 4.3: Global sync time remaining alert in sensor list

The power levels for inputs on the L-band should be between -42dBm and -45dBm (when not observing) and if they exceed these values this will be shown as warning. Low power may mean that the LNA on the receiver are switched OFF or high power might mean AP is pointing at a strong source or ground.

The second power level sensors for the digitisers are the ADC power levels which are measured at the input of the ADC after the gains have been applied. The gains depends on the attenuation levels which can be manually set, but we use separate scripts in setting and refining attenuations. This is important to configure the attenuations so that all antennas have the same output power levels. From the CAM GUI sensor list, filter for adc power i.e. adc.*.power.in. **Figure** 5.2 shows the power levels for each polarisation of m006.

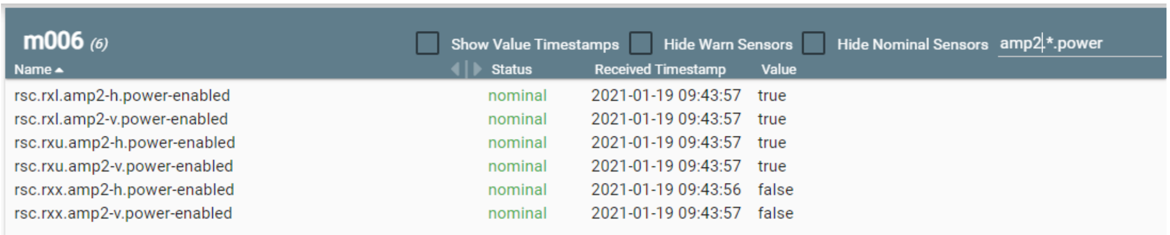


Figure 4.4: Global sync time remaining alert in sensor list

It is recommended that the refine attenuations script should be run after building a new subarray and the set attenuations script be run if there is a change in the signal chain i.e. a Receiver or a Digitiser was replaced.

4.7 Updating config for a replaced digitiser

Digitisers normally have their config files updated before they are handed over to Operations. If a digitiser is replaced, it's serial number needs to be updated in CAM's katconfig configuration files. This will require someone from the operations team to perform github updates. The procedure to do that is shown below.

Note: Swaps done in S-band does not need updates in katconfig, because S-band packetisers don't use serial numbers, but instead use IP addresses according to the receptor number which gets set in the packetiser itself during installation. This is their way of communicating with MeerKAT. Hence the IP address never has to change when S-band packetisers are replaced. There are two different procedures which you can choose to follow:

4.7.1 On the terminal

```
ssh kat@ops.kat.ac.za
cd katconfig/static/antennas/
git checkout master
git pull
git checkout karoo
git pull
git checkout -b update_m0XX_dig_config
```

If the response is:

```
fatal: A branch  named update\_m0XX\_dig_config already exists.
```

then rerun the command, but exclude "-b" this time.

```
git branch
```

This should give: `*update_m0XX_dig_config`

```
ls
vi m0XX.conf
```

(you can also use `nano m0XX.conf` whichever you comfortable with)

Type the letter "i" in order to edit the file. The file format should be for example:

```
digitiser_l = ready:dig-041
```

(we want to change this to the new serial number, i.e 064 for example, to have `digitiser_l = ready:dig-064`)

any digitisers not installed should be of the format:

```
digitiser_x = absent
```

Save the file and exit (press Esc, then type ":wq!")

```
git diff (shows changes you have made, check they are correct)
git add .
git commit -m "Updating (relevant band) digitiser serial no to 0XX on
m0XX"
git push --set-upstream origin update_m0XX_dig_config
```

4.7.2 On Github website

- Go to github: `https://github.com/ska-sa/katconfig`
- Go to the new branch by clicking on the dropdown list on available branches as shown in **Figure** 4.3 (should currently be on the karoo branch) and then selecting the branch name of the new branch created .

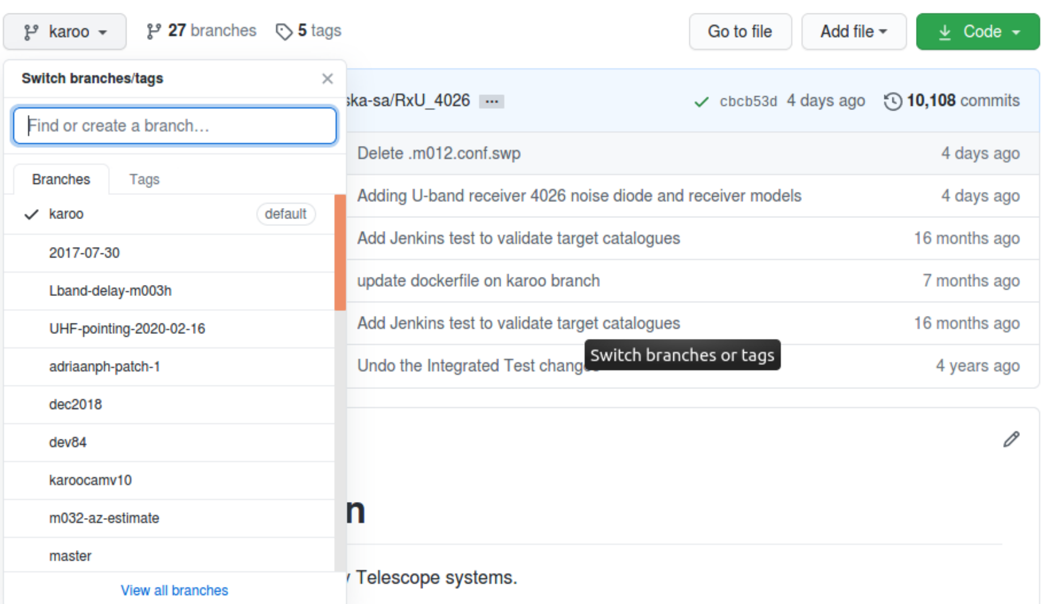


Figure 4.5: Github new branch select

- Create pull request from `update_m0XX_dig_config` (name of branch created) to karoo by clicking on "Pull request" at the top right as shown in **Figure** 4.6.

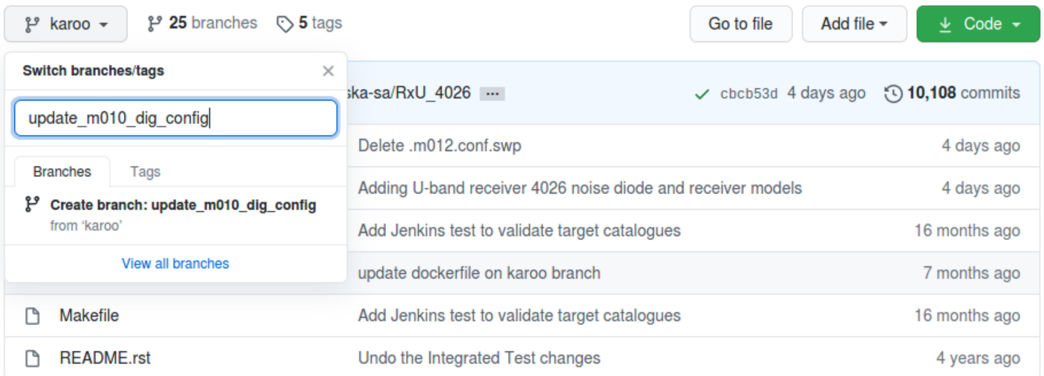


Figure 4.6: Github create pull request

- Make sure `base:karoo` and `compare:update_m0XX_dig_config` is selected. (Also make sure it's only the files you have modified which are part of the pull request - if you did the above steps properly that will be the case).
- Then write a comment stating what you are doing, why (give Jira number if applicable) and make the request out to Pieter Kotze. See an example in **Figure** 4.7.

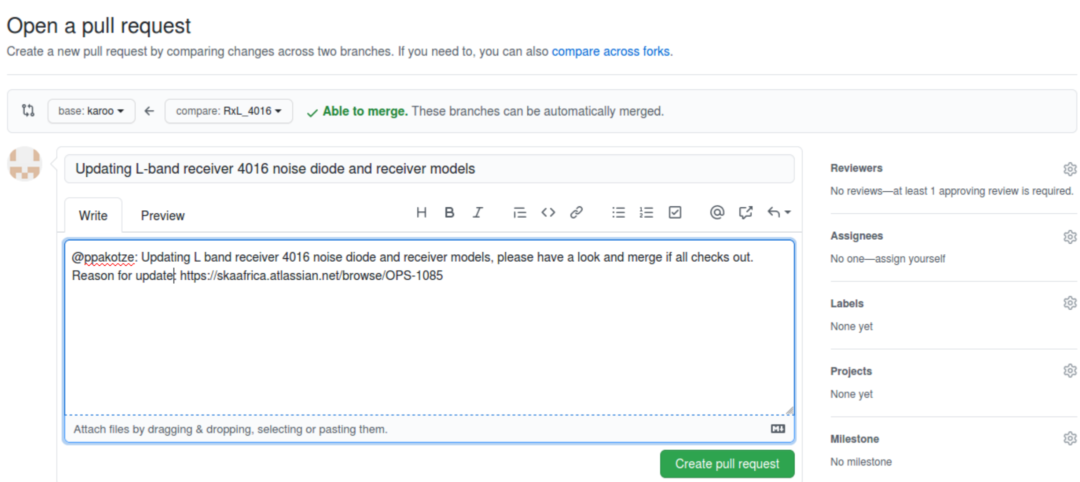


Figure 4.7: Github open pull request

- At the right hand side at the top next to "Reviewers", click on the gear icon. It will drop down a list of reviewers to choose from. Type "ppakotze" (see **Figure** 4.8) in the search field to find Pieter K and then click on Pieter K to request him to approve your pull request (It will show a check mark next to his name, and then under "Reviewers" you will see a yellow dot next to his name).

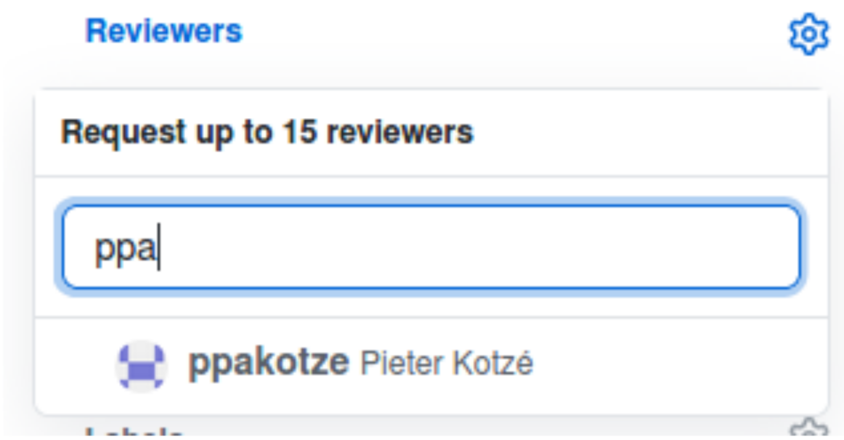


Figure 4.8: Github reviewers dialogue

- Then click on "Create pull request".
  Note: After Pieter K approves the pull request, he usually also merges the pull request, but sometimes you have to do the merging yourself. You do this by going to the pull request after it has been approved, and then click on "Merge pull request" at the bottom and then "Confirm merge"

Chapter 5

Receiver Systems

5.1 Positioning the Receiver Indexer

Receiver Indexer (RI) carousel has four defined positions for each of the receivers it hosts. By selecting the frequency band on CAM's GUI during building of a subarray you are selecting a receiver position mounted on the indexer. Currently we are using L-band and UHF-band for observations and S-band is still being integrated. The raw angles for receivers when they are indexed are:

- L band = 40deg
- X band = 80deg
- U band = 120deg
- S band = 0deg

There is no X−band receiver installed and X−band position is used for parking antennas for the flight arrival and departure on site.

In order to index the receiver to be used for the observation, On the *obs* machine *obs.mkat.karoo.kat.ac.za*, run

```
ipython
Import katuilib
configure_cam('camcam', 'all')
cam.m0xx.req.mode('STOP')
cam.m0xx.req.select_band('l', timeout=60)
cam.m0xx.req.ap_set_indexer_position('l', timeout=60)
```

If you want to use the UHF band for the next observation, replace "l" with "u".

The above-mentioned procedure can be used in instances where the receiver indexer becomes undefined, the *ap.indexer-position* sensor status on the GUI will be error. This can occur when the array is active, oftentimes during a running observation.

5.2 Switching LNAs on and off

The Low Noise Amplifiers (LNAs) do not switch on automatically. They remain off until switched on by the operator. They are switched off automatically when the receiver warms up above 100 K. The 2nd stage amplifiers however, are switched on whenever the cooler is running. The most important sensor to worry about and report when in error is *rsc.rxl.rfe1.temperature* and it must be below 30 K (ideally around 19 K). First, you need to create a configured ipython session by running on `obs` machine:

```
ipython
import katuilib
configure\_cam('camcam', 'all')
```

To turn on the LNAs, run:

```
cam.m0xx.req.rsc_rx(l or u)_lna_h_power('enable')
cam.m0xx.req.rsc_rx(l or u)_lna_v_power('enable')
```

In case you need to power cycle (turn off and then on again) the LNAs, run the following to turn the LNAs off:

```
cam.m0xx.req.rsc_rx(l or u)_lna_h_power('disable')
cam.m0xx.req.rsc_rx(l or u)_lna_v_power('disable')
```

To check in the CAM GUI if the LNA are switched on, select
Main menu > Sensor List > m0XX > rsc.rx(l or u).lna−h.power−enabled nominal true

The image in **Figure** 5.1 shows that LNA for `m006`, L, UHF and S-band receivers are switched on, i.e. enabled.

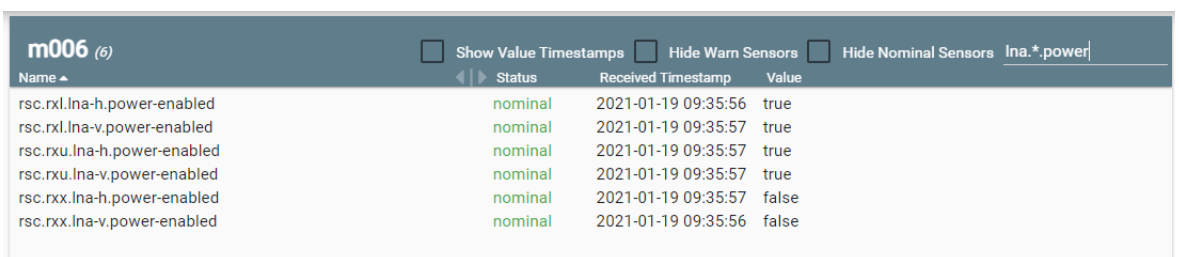


Figure 5.1: Sensor list showing LNA status

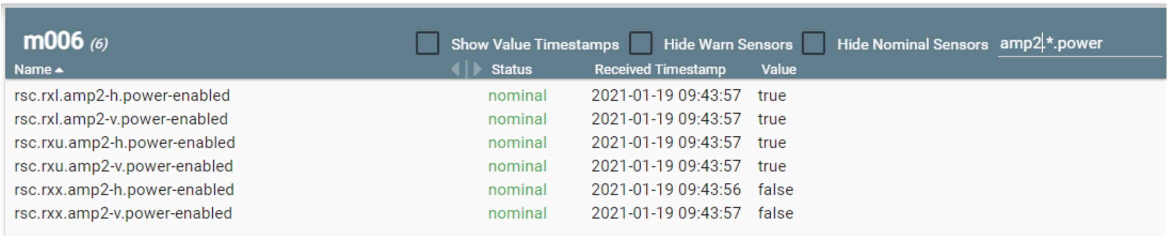
5.3 2nd Stage Amplifiers

To switch them on if they are off, again in the ipython session do the following:

```
cam.m0xx.req.rsc_rx(l)_amp2_h_power('enable')
cam.m0xx.req.rsc_rx(l)_amp2_v_power('enable')
```

To check if they are on, select
Main menu > Sensor List > m0XX > rsc.rxl.amp2−h.power−enabled nominal true

The image in **Figure** 5.2 shows that and stage amplifiers for `m006`, L, UHF and S-band receivers are switched on, i.e. enabled.



m006 (6)

Show Value Timestamps | Hide Warn Sensors | Hide Nominal Sensors | amp2|*.power

Name	Status	Received Timestamp	Value
rsc.rxl.amp2-h.power-enabled	nominal	2021-01-19 09:43:57	true
rsc.rxl.amp2-v.power-enabled	nominal	2021-01-19 09:43:57	true
rsc.rxu.amp2-h.power-enabled	nominal	2021-01-19 09:43:57	true
rsc.rxu.amp2-v.power-enabled	nominal	2021-01-19 09:43:57	true
rsc.rxx.amp2-h.power-enabled	nominal	2021-01-19 09:43:56	false
rsc.rxx.amp2-v.power-enabled	nominal	2021-01-19 09:43:57	false

Figure 5.2: Sensor list showing stage 2 amplifiers

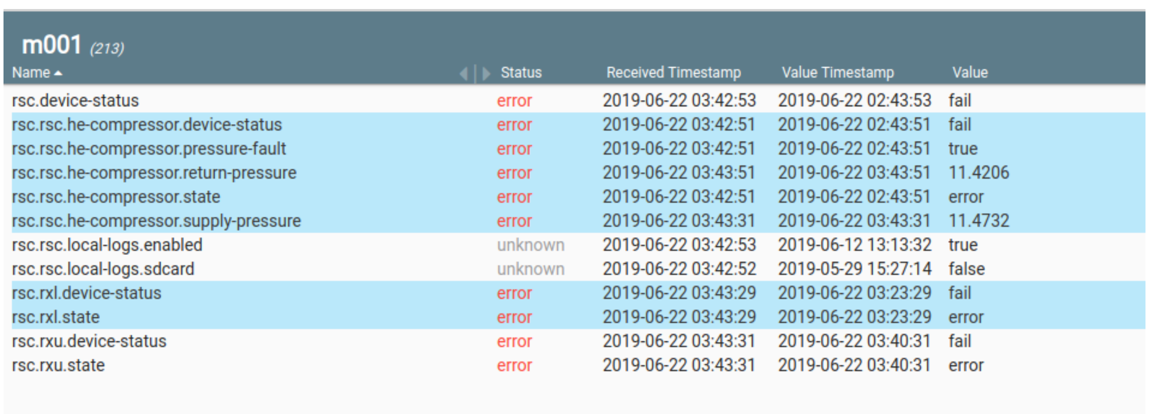
5.4 Helium Compressor

The helium compressor supplies high pressure helium to the cryocoolers of the respective receivers. This means one helium compressor is connected to all working receivers on an AP.

Hence if a fault occurs on the helium compressor, it will shut down causing all receivers (L-band, UHF-band and S-band) on the AP to warm up, making AP unusable for observations. There are usually two types of Helium compressor faults that occur:

5.4.1 Helium compressor pressure fault

Figure 5.3 shows the sensors that go into error when a pressure fault occurs.



m001 (213)

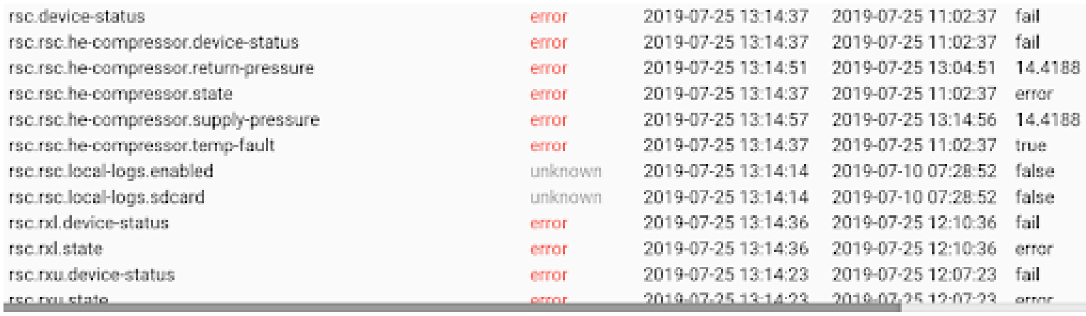
Name	Status	Received Timestamp	Value Timestamp	Value
rsc.device-status	error	2019-06-22 03:42:53	2019-06-22 02:43:53	fail
rsc.rsc.he-compressor.device-status	error	2019-06-22 03:42:51	2019-06-22 02:43:51	fail
rsc.rsc.he-compressor.pressure-fault	error	2019-06-22 03:42:51	2019-06-22 02:43:51	true
rsc.rsc.he-compressor.return-pressure	error	2019-06-22 03:43:51	2019-06-22 03:43:51	11.4206
rsc.rsc.he-compressor.state	error	2019-06-22 03:42:51	2019-06-22 02:43:51	error
rsc.rsc.he-compressor.supply-pressure	error	2019-06-22 03:43:31	2019-06-22 03:43:31	11.4732
rsc.rsc.local-logs.enabled	unknown	2019-06-22 03:42:53	2019-06-12 13:13:32	true
rsc.rsc.local-logs.sdcard	unknown	2019-06-22 03:42:52	2019-05-29 15:27:14	false
rsc.rxl.device-status	error	2019-06-22 03:43:29	2019-06-22 03:23:29	fail
rsc.rxl.state	error	2019-06-22 03:43:29	2019-06-22 03:23:29	error
rsc.rxu.device-status	error	2019-06-22 03:43:31	2019-06-22 03:40:31	fail
rsc.rxu.state	error	2019-06-22 03:43:31	2019-06-22 03:40:31	error

Figure 5.3: Receiver System Hellium presure errors

5.4.2 Helium compressor temperature fault

Figure 5.4 shows the sensors that go into error when a temperature fault occurs. Note the only difference in sensors between the two faults mentioned above is the sensor indicating the type of fault: rsc.rsc.he-compressor-pressure-fault (for pressure faults) rsc.rsc.he-compressor-temp-fault (for temperature faults)

If one of these faults thus occur, report the issue as such (you don't have to also report that the receivers are warm in a separate report, since helium compressor faults automatically cause receivers to warm up).



rsc.device-status	error	2019-07-25 13:14:37	2019-07-25 11:02:37	fail
rsc.rsc.he-compressor.device-status	error	2019-07-25 13:14:37	2019-07-25 11:02:37	fail
rsc.rsc.he-compressor.return-pressure	error	2019-07-25 13:14:51	2019-07-25 13:04:51	14.4188
rsc.rsc.he-compressor.state	error	2019-07-25 13:14:37	2019-07-25 11:02:37	error
rsc.rsc.he-compressor.supply-pressure	error	2019-07-25 13:14:57	2019-07-25 13:14:56	14.4188
rsc.rsc.he-compressor.temp-fault	error	2019-07-25 13:14:37	2019-07-25 11:02:37	true
rsc.rsc.local-logs.enabled	unknown	2019-07-25 13:14:14	2019-07-10 07:28:52	false
rsc.rsc.local-logs.sdcard	unknown	2019-07-25 13:14:14	2019-07-10 07:28:52	false
rsc.rxl.device-status	error	2019-07-25 13:14:36	2019-07-25 12:10:36	fail
rsc.rxl.state	error	2019-07-25 13:14:36	2019-07-25 12:10:36	error
rsc.rxu.device-status	error	2019-07-25 13:14:23	2019-07-25 12:07:23	fail
rsc.rxu.state	error	2019-07-25 13:14:23	2019-07-25 12:07:23	error

Figure 5.4: Receiver System temperature errors

The following helium compressor sensors can be ignored (no action needed) when they go into error as they have no control value over the receivers and thus won't affect observations in any manner:

- *rsc.rsc.he-compressor.temperature1*
- *rsc.rsc.he-compressor.temperature2*

5.5 Receivers system debugging tools

We have debugging tools for the receivers systems which technicians use to predict failures and reporting of the receiver system status. (You can either choose the ***rx_daily*** folder to view the daily receiver reports or the ***rx_monthly*** folder to view the monthly receiver reports). The tool is written in an `IPYTHON` notebook and it is executed daily to provide daily reports.

The following sensors can be ignored (no action needed as they are disabled and thus will show "unknown" status) as they have no control value over the receivers and thus won't affect observations in any manner:

- *rsc.rsc.local-logs.enabled*
- *rsc.rsc.local-logs.sdcard*

Table 5.1: Receiver health by color coding.

Green: Ok											
Yellow : System usable but report maintenance for schedule correction											
Orange : System usable but monitor and report to maintanace if stte persist											
Red : Do not use and report to maitenance immidiately											
Blue: Operator can change state											
	m007	Receiver System	32	rsc	14	rsc.current	1				
	1	Window	1			rsc.voltage	1				

Chapter 6

AP Subsystem

6.1 Motion Profilers

A profiler is needed because the servo loops are high gain loops; large position steps in a high gain system would inevitably lead to large overshoots. The profiler smoothens both the velocity and the acceleration profile. It ensures that the position controller is fed with small position increments only. It also provides velocity and acceleration feed forwards. The acceleration feed forward is enabled only in case of transitions between trajectories. Profiler handling is done automatically by the ACU. External commands from ACU or CAM may include any size of position steps within the operating range.

6.1.1 To check if profilers are on or off

It is required that profiles are always enabled on the AP at all times. But it is advisable that operators check the status of the profilers form time to time.

```
ssh kat@obs.mkat.karoo.kat.ac.za
Ipython
Import katuilib
configure_cam('camcam', 'all')
cam.print_sensors('profiler')
```

6.1.2 To switch profilers off

```
cam.m0XX.req.ap_enable_motion_profiler('elev', 0)
cam.m0XX.req.ap_enable_motion_profiler('azim', 0)
```

6.1.3 To switch profilers on

```
cam.m0XX.req.ap_enable_motion_profiler('elev', 1)
cam.m0XX.req.ap_enable_motion_profiler('azim', 1)
```

6.2 AP Point Error Tiltmeters

The tiltmeter measures the the tilt of the antenna tower and by using a tiltmeter, pointing errors due to non-orthogonality of the azimuth axis and deformations of the azimuth structure because of temperature and constant wind can be compensated to a large extent [TBD]. The tiltmeter shall always be enabled. The status of the tiltmeter can be checked from the ipython session. In the obs machine do the following:

```
ipython
import katuilib
configure_cam('camcam',all)
```

6.2.1 To check the status of the point error tiltmeter sensor

```
cam.print_sensors('point_error_tiltmeter')
```

6.2.2 To enable the tiltmeter sensor

```
cam.m00X.req.ap_enable_point_error_tiltmeter(1)
```

The output will be as follows

```
!ap-enable-point-error-tiltmeter ok
```

6.2.3 To enable all antennas

```
cam.ants.req.ap_enable_point_error_tiltmeter(1)
```

6.2.4 To disable the tiltmeter sensor

```
cam.m00X.req.ap_enable_point_error_tiltmeter(0)
```

The output will be as follows

```
!ap-enable-point-error-tiltmeter ok
```

6.3 Test observations scripts

6.3.1 SE tilt sensor measurements observations

The purpose of the tilt measurements is to look at calibration tilt sensors and also to calculate tilt sensor offset parameters and compare these with set values. From a single test the calculated tilt sensor parameters did not match that closely to the set values. We were hoping to take many measurements so that we could see from the average of a number of low wind measurements if the calculated value matched the set values, especially for receptors where the calibration had recently been checked. The next step was going to be to get to a point whereby we could automatically write the calculated values to the ACU to update the values. We have not reached that point yet.

6.3.2 Ap.rate_test.py

This script rotates the AP at elevation 20 tilt measurements. This script is no longer used and was applicable during acceptance testing and commissioning.

Ap.rate_test.py fails due to timeout

cause: the timeout happens because some of the antennas become unresponsive during the measurement.

consequences: currently this causes the script to terminate with a messy error message − fortunately it doesn't mess up the telescope state (as it used to do earlier in the year). when this happens the first 1/2 of the script gets completed correctly, just the second 1/2 is interrupted. The second 1/2 is essentially a repeat of the first, so there's still useful data here.

recommendations: the current behaviour is messy but not intolerable. the script should be made more robust against this, to continue with the remaining antennas.

6.3.3 Reporting AP Struct Tilt x/y in error

Risk

- Pointing offsets

Procedure

- Inspect the status of tilt sensors in the GUI sensor list

  $$\text{ap.struct}-\text{tilt}-\text{x}\ \text{error}\ 2019-07-12\quad 14:59:06$$
  $$\text{ap.struct}-\text{tilt}-\text{y}\ \text{error}\ 2019-07-12\quad 14:59:16-190.64$$

  When reporting this error include the following in the JIRA to determine the course and action.

  - Check the ap temp − plot sensorgraph
  - Check ap motion − plot azim and elev actual sensors
  - Is x/y tilt getting larger(correction value)? (Plot struct tilt x/y on sensorgraph)
  - Or is it a sudden jump to error?
  - Checking these trends below will answer the question as to whether this is a matter of calibrating with correct values or is it a structural problem or not.

- Report the JIRA to site AP teachnicians

Chapter 7

Engineering and Maintenance

Every Wednesday is reserved for maintenance, upgraded and deployment of new software and hardware. This time is usually given to engineers and technicians to sort out issues and faults that have been reported by Telescope Operators in Userlogs and Jiras. There are still observations that are carried out depending on the number of antennas. But this day is mainly used for integration testing and bug fixing.

7.1 Flights

On Wednesday morning, a flight takes off from Cape Town International Airport with a number of engineers who go to site to do different types of tasks like hardware installation, testing, bug fixes etc on site. For days and times when there will be a flight arriving on site, it is important to stow the receptors away from the flight path. The flight company usually sends the ETA for the destination at Losberg and the flight arrival and departures are marked on the site calendar. The procedure to follow is to build a subarray and then use the following schedule block (SB), which must be verified on the GUI and run as an observation, ensure that receptors are pointing all at elevation 18 deg.
In ipython session do the following commands:

```
configure_obs()
obs.sb.new(owner="Operator")
obs.sb.type = katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.description = "Lower APs for Flight arrival and departure"
obs.sb.instruction_set= "run-obs-script /home/kat/katsdpscripts/utility/flight_stow.py"
obs.sb.to_defined()
obs.sb.to_approved()
obs.sb.unload()
```

You can run the observation like any other observation except there is no need to include other resources like cbf and sdp in the subarray.

7.2 ComRAD

Use this tool to track Execujet flights every Wednesday so to stow/unstow antennas to protect the receivers[4]. `http://comradgis.kat.ac.za:8000/login`
You can create your own username and password.

Chapter 8

Telescope Control Procedure

8.1 The Site Calendar

The site calendar is used to schedule all observations and testing that is carried out on meerKAT telescope. The operator will typically follow the instructions that are scheduled on this calendar to do global sync, calibration, stowing the antennas for flights, build subarrays etc. The operator will be required to do the following as per the calendar entries:

- Verify that you are on AR1_site calendar, see **Figure** 8.1 below for depiction of the site calendar.
- If no schedule block (SB) is assigned, then proceed to the section below.

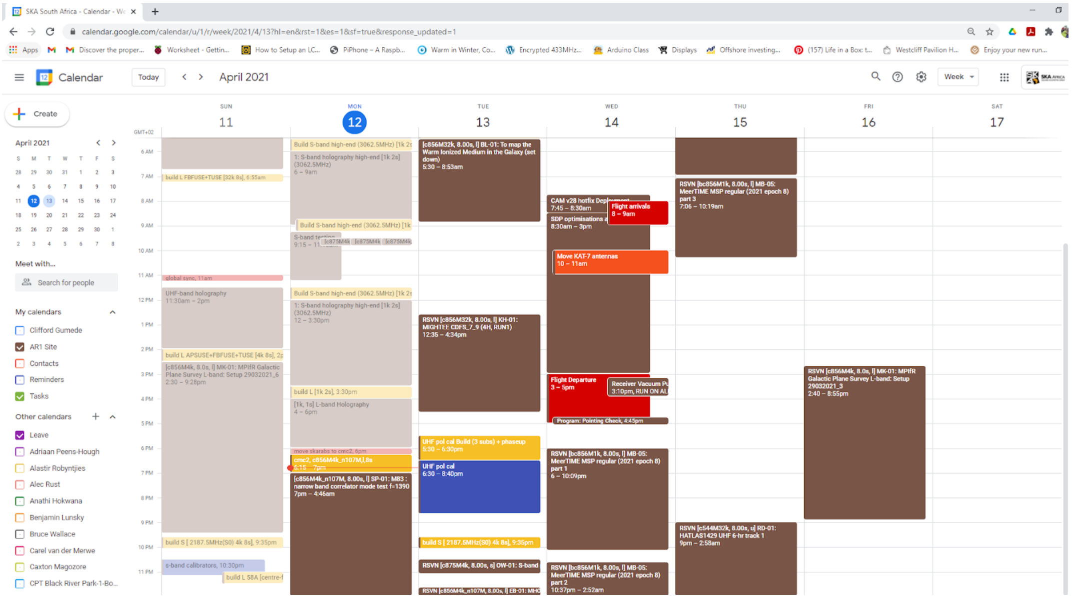


Figure 8.1: Google calendar for site activities

8.2 Create a New Schedule Block (SB)

Connect to obs machine via

```
ssh kat@obs.mkat.karoo.ac.za
ipython
Import katuilib
configure_obs()
```

Paste the schedule block details as shown in the image below in **Figure** 8.2 and run from the calendar event (observation).

Figure 8.2: SB pasted on ipython sesion

The commands above when ran in this ipython session will create a new SB in the GUI as shown in **Figure** 8.3.

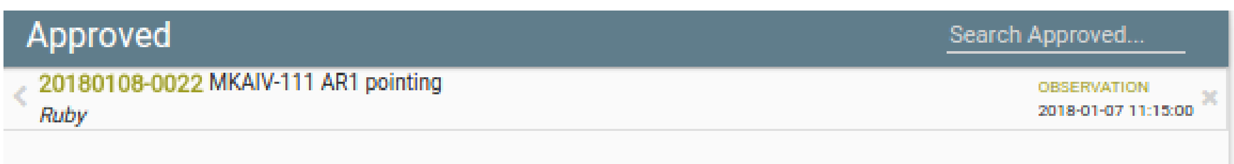


Figure 8.3: CAM GUI approved SB

8.3 Retrieve Old Schedule Blocks

If you need to find a schedule block (SB) that is no longer found in the complete SB history in the GUI as shown in **Figure** 8.4 below.

Figure 8.4: Completed SB search

Connect to the obs machin by ssh and run the commands as shown in **Figure** 8.5 .

```
ssh kat@obs.mkat.karoo.kat.ac.za
```

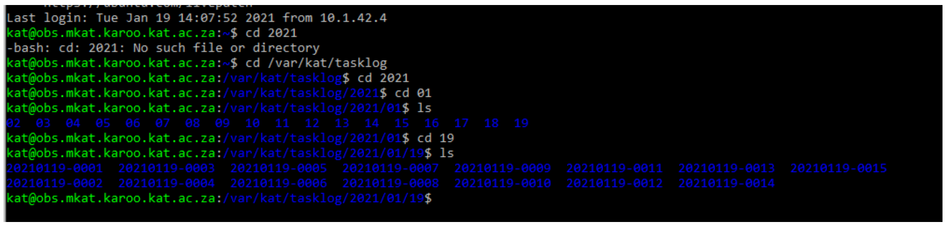


Figure 8.5: SB search in the obs machine

Select the schedule block number that you need. You can view the contents of the schedule block by running the following commands on the ipython session:

```
configure_obs()
obs.sb.new_clone(sb_number'
obs.sb.load('new_sb_number' )
```

If you do not know what the SB number is but know when the observation ran, run the following commands to display list of activities as shown in **Figure** 8.6

```
ssh kat@portal.mkat.karoo.kat.ac.za
cd /var/kat/log$
less activity.2020-05-27.log  (you can choose the date of the log you want to view)
```

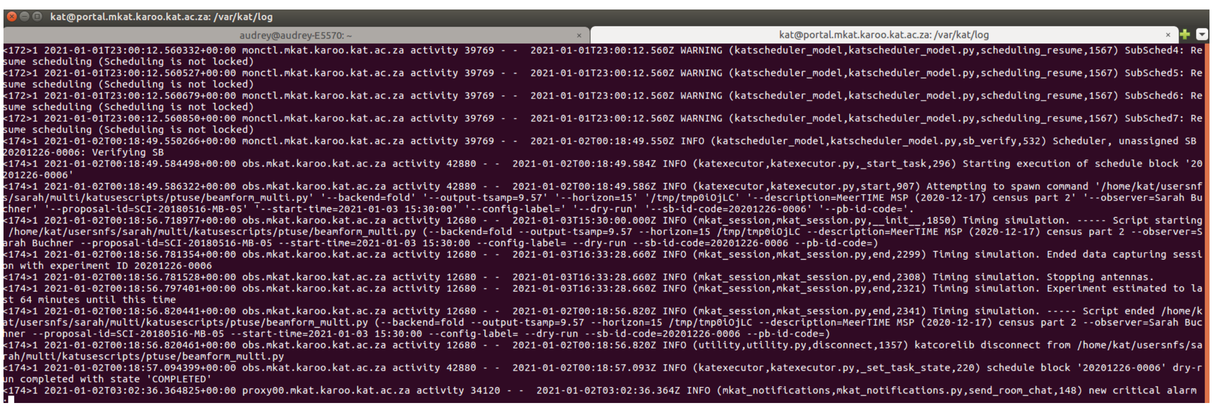


Figure 8.6: Open SB log file

Zoom in on the screenshot to see schedule block numbers that ran/dry-ran on your chosen date.

8.4 Build a subarray (SA) suitable for the SB (observation)

Select and load the desired resources from the free resources. These are: Receptors (AP) The correlator (cbf_x) for *cmc1* or cbf_dev_x for *cmc2* Science Processor (sdp_x) Note that the Correlator and Science Processor proxy numbers denoted by _x must match the subarray number (1 to 4) Select the data user product (List provided in drop down menu)

Select desired frequency band (l,u,s or x) Select desired dump rate (list provided in drop down menu) If the dump rate was not specified for the SB (observation) leave the default setting as is. Select APSUSE, FBFUSE, PTUSE and TUSE and if neededActivate the subarray. When the array is active, it will appear green in **Figure** 12.11

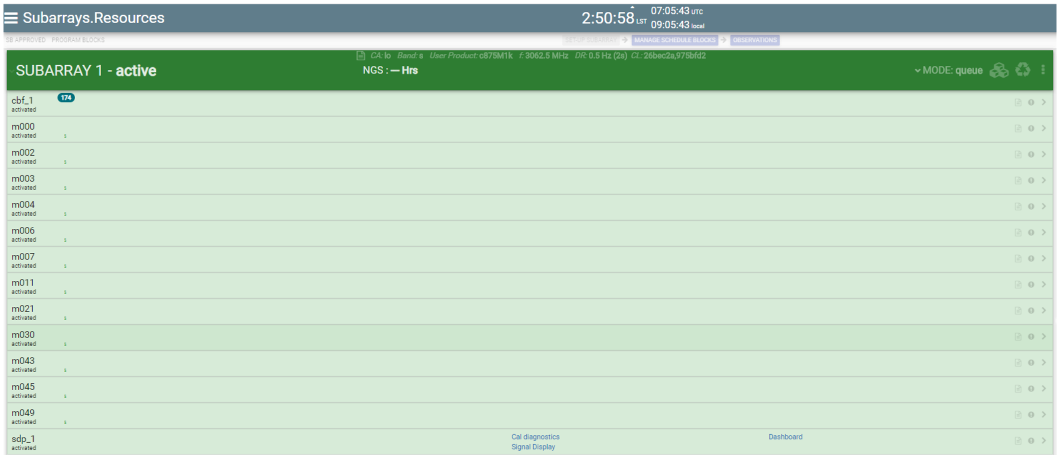


Figure 8.7: Active subarray with `cbf_1` and `sdp_1` resources

Verify the SB by clicking on the far right dots of sb, see **Figure** 8.8.

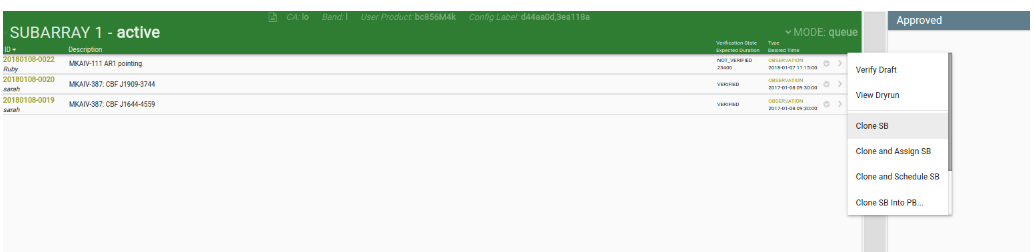


Figure 8.8: Verifying a SB

- While verifying, check the dry run,
- run the Schedule Block
- Schedule the SB
- Execute the schedule
- Scheduler modes should be in
  - Manual - manual control over observation start-ups
  - Queue - observations will automatically run as per queue

8.5 Alternative SA building using IPython session for debugging

#Creating a small subarray using python session

```
f=configure_subarray('test', 1)  ## build  a subarray (this is after the subarray is activated in the G
f.sub.sensor.number_ants.get_value()  ## to check if the subarray is active
f.ants.req.mode("STOP")
f.ants.req.target_azel(200,15)
f.ants.req.mode("POINT")
```

This example illustrate how to run a subarray manually using CAM commands

8.6 Converting type of SB from 'MANUAL' to 'OBSERVATION'

There are times when instruction sets could generate schedule blocks on the GUI with the observation type as MANUAL as in **Figure** 8.9 :

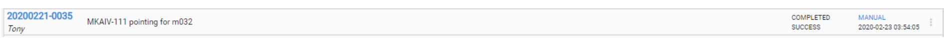


Figure 8.9: Manual SB

If this happens, the dry run and progress log will be empty. The observation type can be changed on the GUI by following the steps below:

First assign the SB to the subarray as shown in **Figure** 8.10

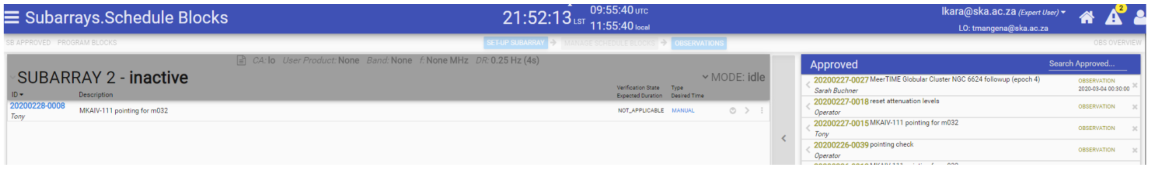


Figure 8.10: A SB assigned to a subarray_2

Click on 'SB APPROVED', see **Figure** 8.11

Figure 8.11: Approved SB

Choose 'Edit' as shown in **Figure** 8.12

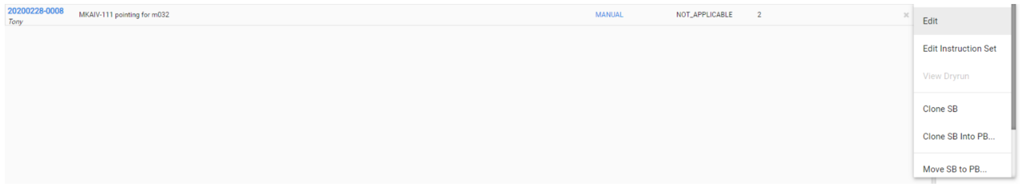


Figure 8.12: Edit the approved SB

Choose ‘OBSERVATION’ in the drop-down menu under ‘TYPE’

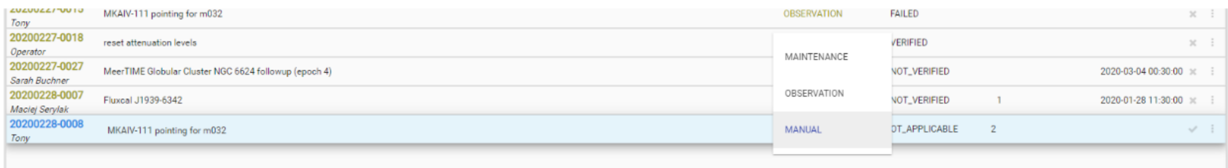


Figure 8.13: Save SB as observation type

Save that option by clicking on the tick

Figure 8.14: Save the edited SB

If you verify the SB this time, it should give you a dry run.

8.7 Reset Attenuations

This script can be run immediately after building an array when antennas have been out for maintenance or when changing between bands.

The power level into the digitiser (*rfcu.hpol.rf.power.in*) cannot be adjusted as this is a measure of the signal power produced by the receiver. Gain adjustment inside the receiver is not possible. If this power level is in error, operations should investigate the health of the receiver. Operations can, and should, however adjust the power level that enters the ADC (*adc.hpol.rf.power.in*). If this sensor is in ERROR, operations should firstly verify the state of the RF power into the digitiser (*rfcu.hpol.rf.power.in*) to ensure that it is not in ERROR. If not the gain of the digitiser should be determined to ensure that it is within limits. Once it has been verified that the gain is within limits, the digitiser attenuation must be decreased until the power level into the ADC is within limits. There is no need for all the attenuators to be set to the same value. The reason for the attenuators is to adjust each receptor values individually. I therefore expect to see variations in attenuation across the array.

The second thing to remember is that the digitiser has a lot of dynamic range which means that there is very little chance of any saturation happening in the digitiser. You should therefore not unnecessarily add attenuation. It would be better to reduce attenuation (more gain) than add attenuation.

Attenuation levels should always be set to appropriate levels depending on the target brightness. For very bright sources, attenuation levels are set at 31. For most of our observations attenuation levels are set 6 for L-band.

Source:

```
kat@obs.mkat.karoo.kat.ac.za:~$ less STANDARD_SCHEDULE_BLOCKS.TXT
```

In the obs machine, you must login ipython session and run:

```
configure_obs()
obs.sb.new(owner='SeanPassmoor')
obs.sb.type=katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.description='Reset  attenuation levels'
obs.sb.antenna_spec='available'
obs.sb.instruction_set="run-obs-script /home/kat/katsdpscripts/utility/set_attenuation.py /home/kat/kat
obs.sb.to_defined()
obs.sb.to_approved()
```

You may consult with the AOD if the power levels do not look levelled, or look suspicious. There is a script below that you can run to refine attenuations found in the same folder above.

On the terminal, run:

```
ssh kat@obs.mkat.karoo.kat.ac.za
ipython
```

and paste the following SB:

```
configure_obs()
obs.sb.new(owner="SeanPassmoor")
obs.sb.type = katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.description = "Refine Attenuation"
obs.sb.antenna_spec="available"
obs.sb.instruction_set= "run-obs-script /home/kat/katsdpscripts/utility/refine_attenuation.py"
obs.sb.proposal_id='CAL-Attenuate'
obs.sb.to_defined()
obs.sb.to_approved()
obs.sb.unload()
```

8.8 Setting digitiser attenuation via ipython session

On the *obs* machine/server. Configure a cam object:

```
ssh kat@obs.mkat.karoo.kat.ac.za
ipython
import katuilib
configure_cam('camcam','all')
```

Check attenuation:

```
cam.print_sensors('attenuation')
```

The steps below must be done before building a subarray but it is not advisable to set attenuations manually as there is a script that was developed that has automated this process. These steps below can be used for test purposes only.

8.8.1 To attenuate all digitiser

```
cam.ants.req.dig_attenuation('h',6)
cam.ants.req.dig_attenuation('v',6)
```

This will apply attenuation of 6 to both vertical (v) and horizontal(h) polarisations of all antennas. This must not be done under any circumstances unless debugging the system.

8.8.2 To attenuate a single digitiser

```
cam.m0xx.req.dig_attenuation('h',6)
cam.m0xx.req.dig_attenuation('v',6)
```

8.9 Delay Calibration and Phase-ups

This is done to ensure that the signal has a phase difference of zero. Solutions close to zero are acceptable but monitor the waterfall plot as indicated below if unsure. To find a procedure on how access signal displays and plot waterfall displays go, go to the section called “Monitor an active observation” in this manual.

Risk Incorrect delay solutions or out of phase signal data on some baselines

Source:

```
kat@obs.mkat.karoo.kat.ac.za:~$ less STANDARD_SCHEDULE_BLOCKS.TXT
```

calibrate_delays and *bf_phaseup* have been updated to use the new ’default’ gain feature in CBF. It is no longer necessary to specify `fft-shift` and `f-engine-gain` options in the instruction set. Run a delay cal immediately after every new subarray build Monitor the progress output of the script and waterfall plot of all baselines

8.9.1 Delay calibration SB

The following SB is the Standard delay calibration. This can be run for 1K, 4K, 32K and in UHF-band and L-band.

```
configure_obs()
obs.sb.new(owner='Operator')
obs.sb.type=katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.controlled_resources_spec='cbf,sdp'
obs.sb.description='Delaycal'
obs.sb.instruction_set="run-obs-script /home/kat/katsdpscripts/observation/calibrate_delays.py '/home/k
obs.sb.proposal_id='CAL-20200106-OP-02'
obs.sb.notes='Delay calibration.'
obs.sb.to_defined()
obs.sb.to_approved()
obs.sb.unload()
```

8.9.2 Phaseup calibration SB

The following SB is the Standard phase up with bandpass flattening. This can be run for 1K, 4K, 32K and in UHF-band and L-band.

```
configure_obs()
obs.sb.new(owner='Operator')
obs.sb.type=katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.controlled_resources_spec='cbf,sdp'
obs.sb.description='Phase up with flatten bandpass'
obs.sb.instruction_set="run-obs-script /home/kat/katsdpscripts/observation/bf_phaseup.py '/home/kat/kat
obs.sb.proposal_id='CAL-20200106-OP-03'
obs.sb.notes='Phase up with bandpass flattening.'
obs.sb.to_defined()
obs.sb.to_approved()
obs.sb.unload()
```

Accepted waterfall plots - in **Figure** 8.15, looking from the top of each y-axis baseline block (time) Green colour means there is no phase difference - this is good. Other colours are acceptable too. These are minor differences.

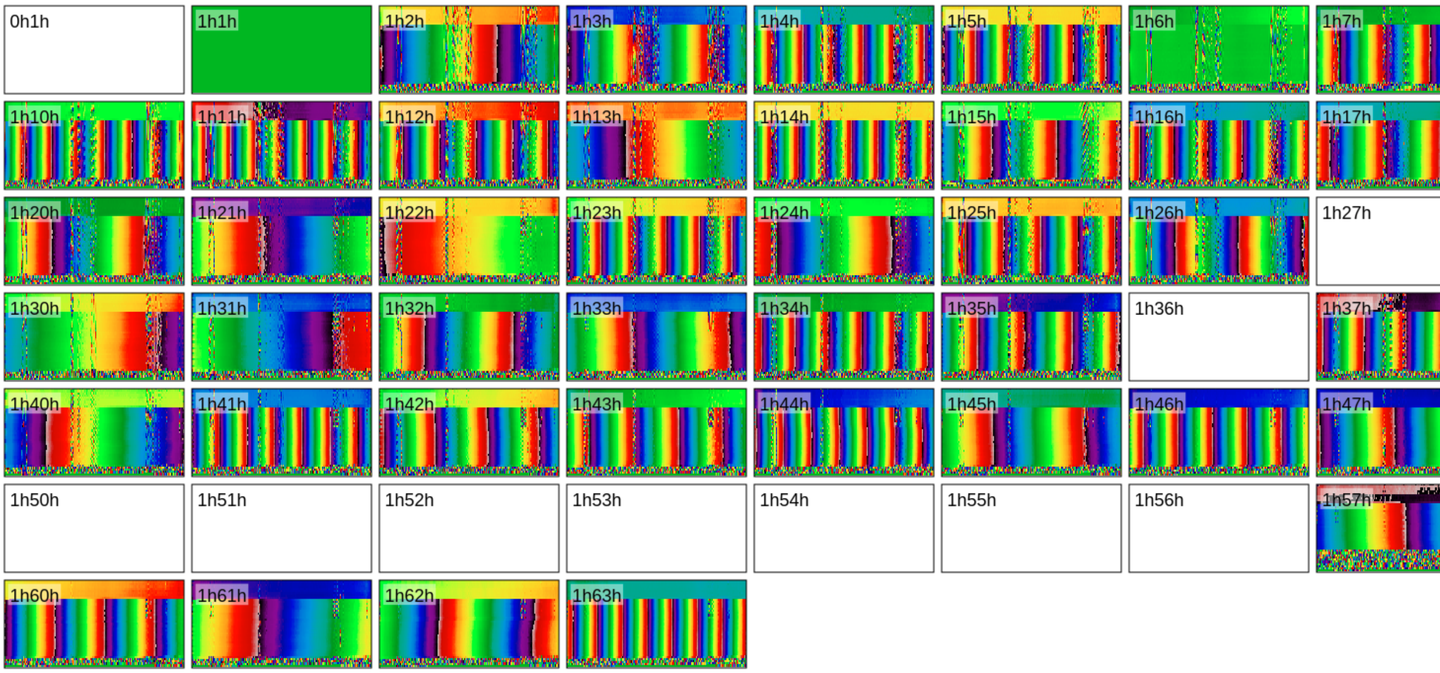


Figure 8.15: Acceptable waterfall plot

- For **all observation types**, noisy output/incoherent signal output is not acceptable, mark that antenna faulty.
- For **imaging observations**, phase wrapping can be corrected by the Astronomer, do not mark that antenna faulty.
- For **Holography and Pulsar observations**, any large delay solution values, mark the antenna faulty.

8.9.3 Delay Cal Script Fails:

Delay solutions not found:

```
2020-08-31 16:32:39.386Z WARNING - m063h:  0.000 ns, delay fit failed (all its
data probably flagged)
2020-08-31 16:32:39.387Z WARNING - m063v:  0.000 ns, delay fit failed (all its
data probably flagged)
```

- Check *kat@portal.mkat.karoo.kat.ac.za:/var/kat/log$* (specific antenna log) - to see if the antenna misbehaved during the delay cal. e.g ap failures/proxy not in STOP when it should be.

- Or check *kat@portal.mkat.karoo.kat.ac.za:/var/kat/log$* for ***kat.cbfmon_1*** logs to see around that time that the array was built if there were any HMC or sync errors or any other cbf error regarding a F-host/X-host connected to that failed antenna.
- Check cal pipeline logs via mesos in the GUI

8.10 What could go wrong?

8.10.1 Failed to select band while building array

- One or more receptors failed to select observing band. The following error can be seen when building an array from the subarray logs:
  `'select_band#012 raise faults. SelectBandError(failed)#012SelectBandError: Failed to select_band on receptors [m0xx]'`
- Wait for the building process to end, then press the retry button on the GUI.
- If the above step fails, check the *'indexer raw position'* sensor in sensor list (see **Figure** 8.16) for that receptor (e.g L band = 40 deg)

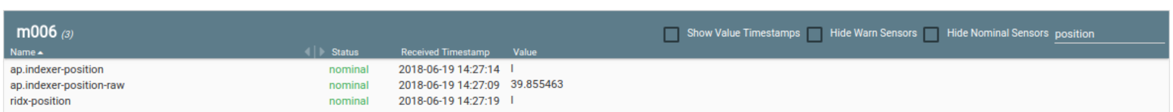


Figure 8.16: CAM GUI sensor list to check indexer position

  If not on band, ssh into the obs machine as previously done.

```
ipython
import katuilib
configure_cam('camcam', 'all')
cam.m0xx.req.mode('STOP')
cam.m0xx.req.select_band('l', timeout=60) or 'u' for UHF
cam.m0xx.req.ap_set_indexer_position('l') or 'u' for UHF
```

- If the above step fails, check if the digitiser for the band in use is marked ready as shown in **Figure** 8.17. If not refer to 'mark digitiser ready' section on this document, or follow the steps on the Mark Digitiser Absent and/or Ready procedure.

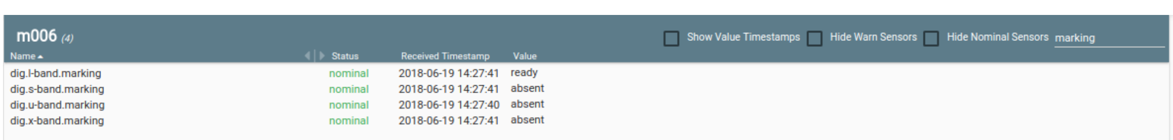


Figure 8.17: CAM GUI sensor list to check digitiser marking

- If unresolved, check if the digitiser serial number is the right one. To do this refer to the 'Updating config for a replaced digitiser' section on the document or follow the Checking digitiser serial numbers procedure. Should the wrong digitiser be marked ready or the issue remain unresolved contact CAM Support.

8.10.2 Multiple F-hosts in error after subarray build

- To identify the disabled f-hosts
  - Once the cbf is activated check the cbfmon plot in the GUI
  - The antennas appears as a green (zero) blocks in waterfall plot
  - You will see the signal for the affected antenna drop in the time series plot
  - You will also notice no ingest data on the signal display as shown in **Figure** 8.18. Type `flagcount` to display the figure below.

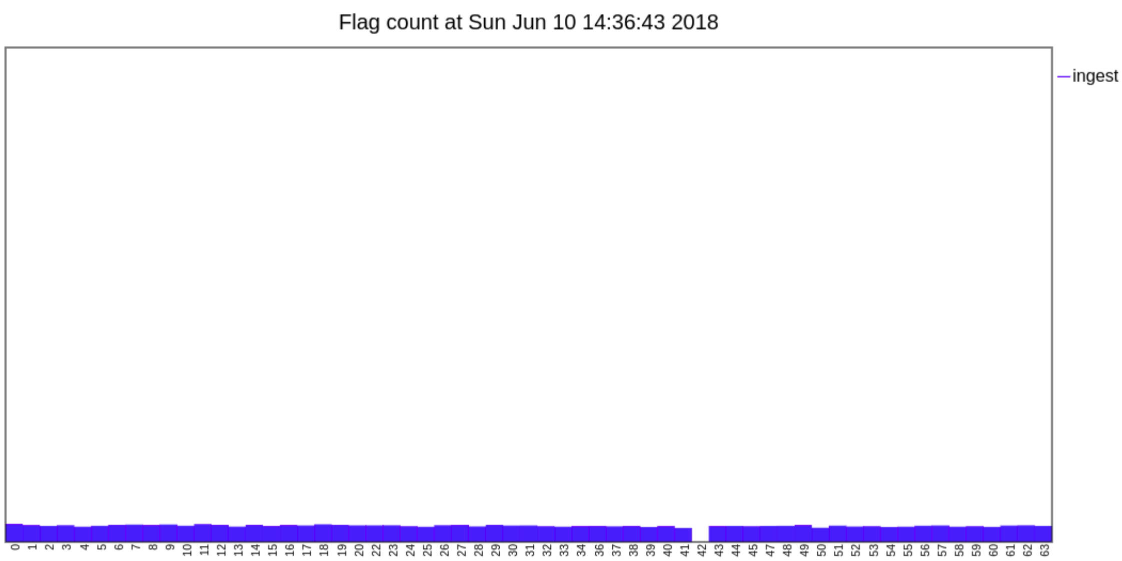


Figure 8.18: SDP flag count plot with one F-host disabled

- CBF should automatically disable a faulty F-HOST.
- To verify cbf health, go to:
  `http://cbf-nuc2.cpt.kat.ac.za/cmc1/array_1.wide`
- You can also check the sensors by looking at the CBFMON on the sensor list (hide nominal - this takes a while to load)
- If multiple F-hosts are in error, break the array and build a new one [Remember to double check over a few minutes to see if the errors do not disappear by themselves.]
- The other option is to check the health of CBF via the CAM GUI as shown in **Figure** 8.19 and **Figure** 8.20

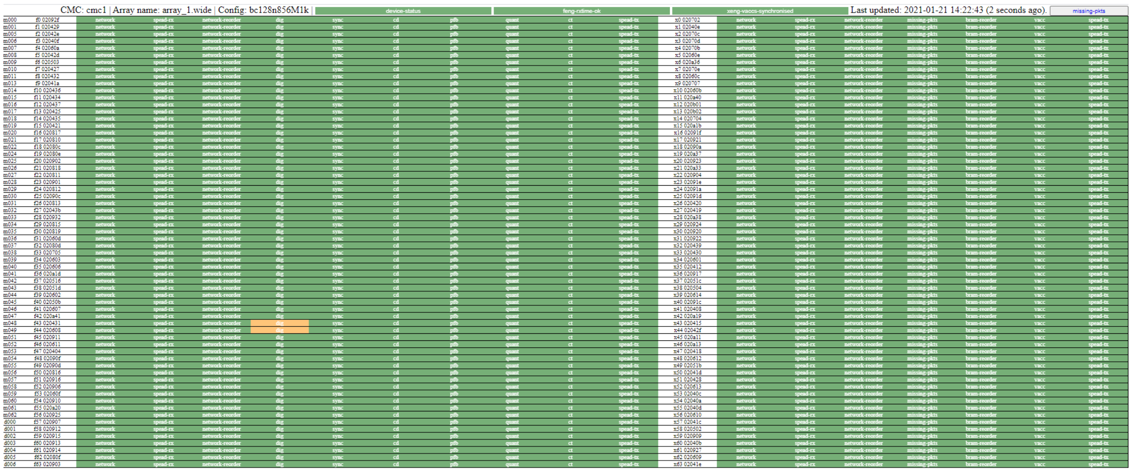


Figure 8.19: CBF health dashboard

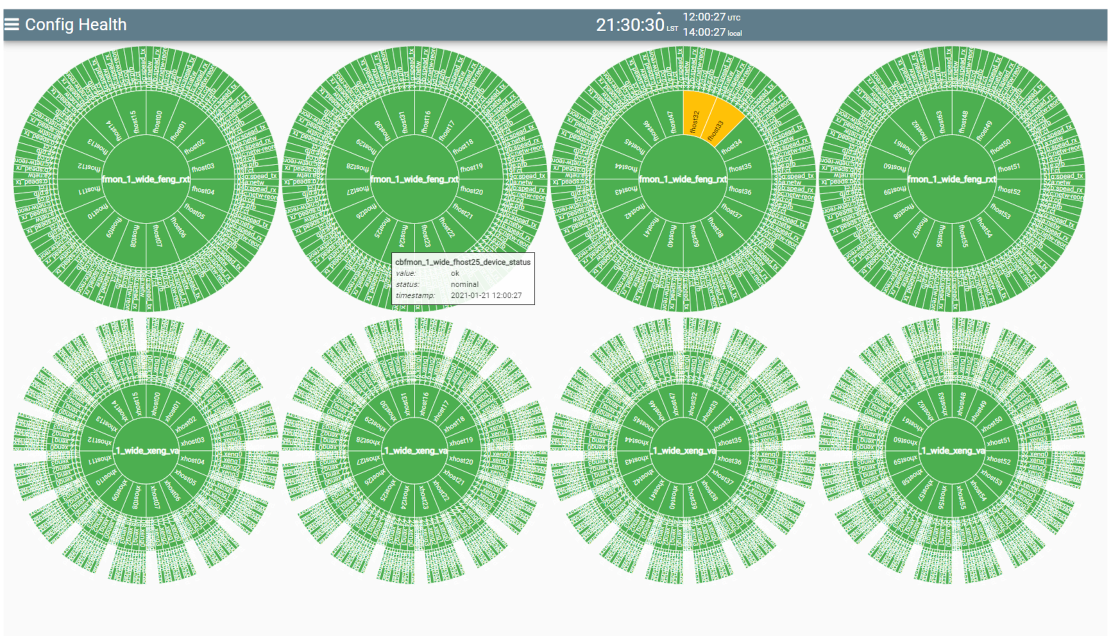


Figure 8.20: CBF health pie charts

8.10.3 Marking SKARAB Board on Standby

If these f-host errors are mapped to one particular bad skarab board call cbf to disable the board. In obs machine type the following commands to check on CMC2:

```
kcpcmd -t 30 -s 10.103.254.3:7147 array-list
kcpcmd -t 30 -s 10.103.254.3:7147 resource-list|grep "up$" | wc -l
kcpcmd -t 10 -s 10.103.254.3:7147 resource-mark skarab020408-01 standby
```

8.10.4 No Descriptor Errors

- After a subarray is built run an observation to check that descriptors are being received.

- It is a good idea to run a delay cal with a long "tail" `--verify-duration=300`, this will return to the source for 300s after the delays are found

8.10.5 Diagnostics

- When running the observation it is useful to have two signals displays (windows) open
  - Grafana
  - Go to the GUI, sensor list and select sdp_1 , Monitor descriptors in sensor list (search 'desc')
  - The no descriptor count should not be rising
  - There are four ingest nodes - each has 1/4 of the band
- Example of "**no descriptor**" error
  - No descriptor block shows orange
  - Part of the band is missing, see **Figure** 8.21 and **Figure** 8.22.
  - The no descriptor heap counts keeps rising as shown in **Figure** 8.23

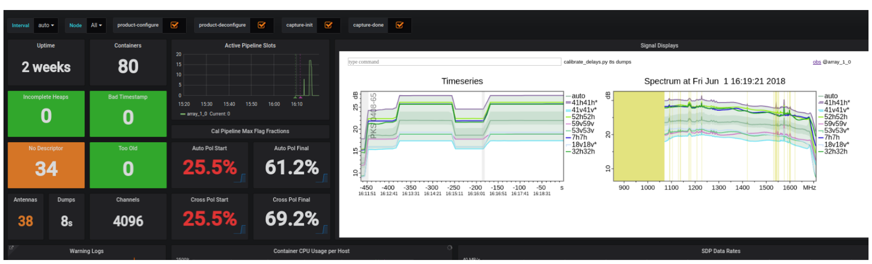


Figure 8.21: SDP "no descriptors" error on Grafana

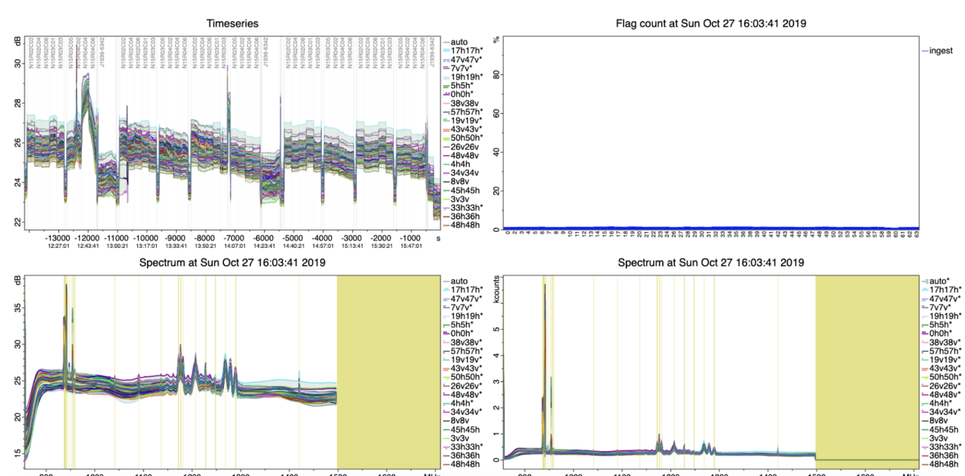


Figure 8.22: Part of band missing due to "no descriptors" error

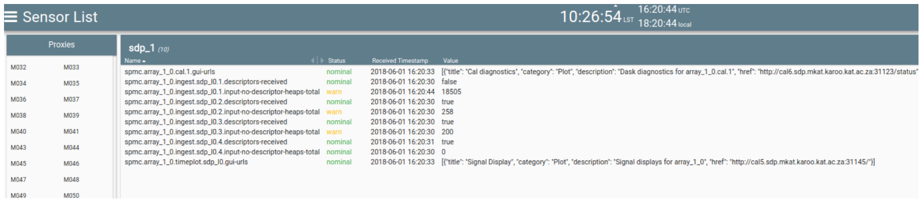


Figure 8.23: SDP sensor list with no-descriptor count increasing

8.10.6 How to fix this

Step 1 Stop the current observation.

- Check the number of "no descriptor" errors, check x-host health, check leaf switches, and if they are okay continue to step 2.

Step 2 Tear down the array for critical/urgent observations to continue.

- This should leave the failed subarray as a zombie within SDP for debugging, while continuing operations.
- To diagnose if this is an SDP problem:
  - ☐ Find the time at which the ingest process stopped working: this will show up as a sudden drop in the SDP input data rate.
  - ☐ Then check SDP's (ingest) logs for a message like this at around the same time:
    - `2019-10-27 15:59:58.558 ing4.sdp.mkat.karoo.kat.ac.za ingest.sdp_l0.4: array_1_wide_0 worker thread blocked by full ringbuffer on heap 8467887039`

Step 3 After following the above steps, rebuild the subarray and monitor the leaf switches health using the following command line on obs machine :

```
ssh kat@obs.mkat.karoo.kat.ac.za

ssh kat@obs.mkat.karoo.kat.ac.za: ./usersnfs/cbf_support/display_switch_rates.py -d leaves
```

8.10.7 CBF Data Capture Failed

See the CBF data capture error the progress out put
`http://10.97.1.13:8081/tailtask/20180612-0012/progress`

- This error on the cbf side is caused by a clash in resources (where some skarabs are being deprogrammed in the lower cbf levels by a non lead operator)

- Break the array

```
ssh telnet cmc1.cbf.mkat.karoo.kat.ac.za 7147
?resource-list (to see which resources are available)
```

- Try building again, if this error persists over several retries contact cbf to release resources
- There is no fix in place to stop interruptions such as these - look at JIRA below `https://skaafrica.atlassian.net/browse/MKAIV-1169?filter=-2`

8.10.8 SDP Capture Done Failure

Interim SOP for CAM subarray-product-ids that end up causing sdp proxy to go in error while running scripts or rebuilding subarray.

- Check via the observation script logs which is visible with a warning followed by an error message.
- Check via CAM sensorlist for cam data proxy, sdp_1; filter product−ids value. `http://portal.mkat.karoo.kat.ac.za/katgui/sensor-list?component=sdp\_1`

  If it shows more than 2 values for that same subarray, then it's a clear indication to teardown and rebuild the subarray.

  Example, on subarray_1, you will see these values: `array_1_wide_0, array_1_wide_1`

  `Subarray-product-ids nominal 2019-11-28 13:52:20 array_1_wide_0, array_1_wide_1`
- Tear-down and rebuild the same subarray. This should clear the error
- Re-check via CAM sensor-list for sdp_1; use filter: subarray-product-ids, and it should show only one value as shown in **Figure** 8.24.

  `Subarray-product-ids nominal 2019-11-28 13:52:20 array_1_wide_0`

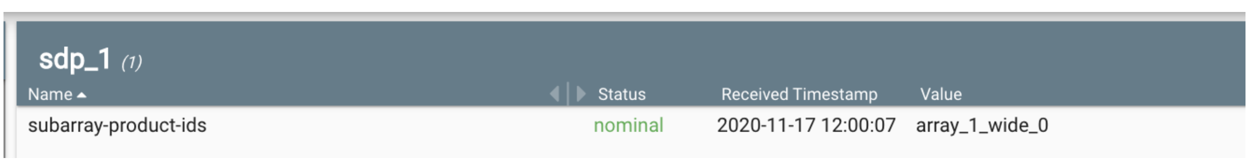


Figure 8.24: Subarray product-ids

Note: Only use restarting the sdp proxies via CAM as a last resort or when instructed.

8.10.9 Vaccs Lost Sync

When vaccs lose sync an X-engine becomes disabled

- If during an observation, data on some band suddenly disappears as shown in . Check the signal displays as shown **Figure** 8.25
- Run the check cbf health script in the obs machine

  ```
  kat@obs.mkat.karoo.kat.ac.za:~$ ./katsdpscripts/utility/check_cbf_skarab.py  --strategy=detail
  ```
- If the vaccs have lost sync, kill the array and build a new one
- Write a JIRA with a sensorgraph as shown in **Figure** 8.26

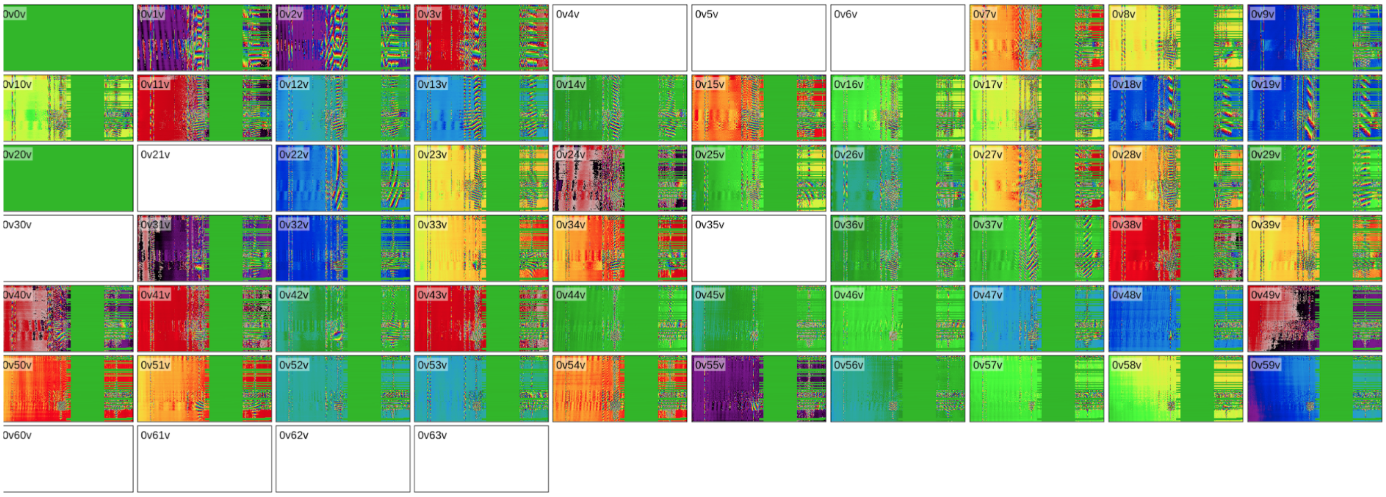


Figure 8.25: Data loss due to vaccs sync error

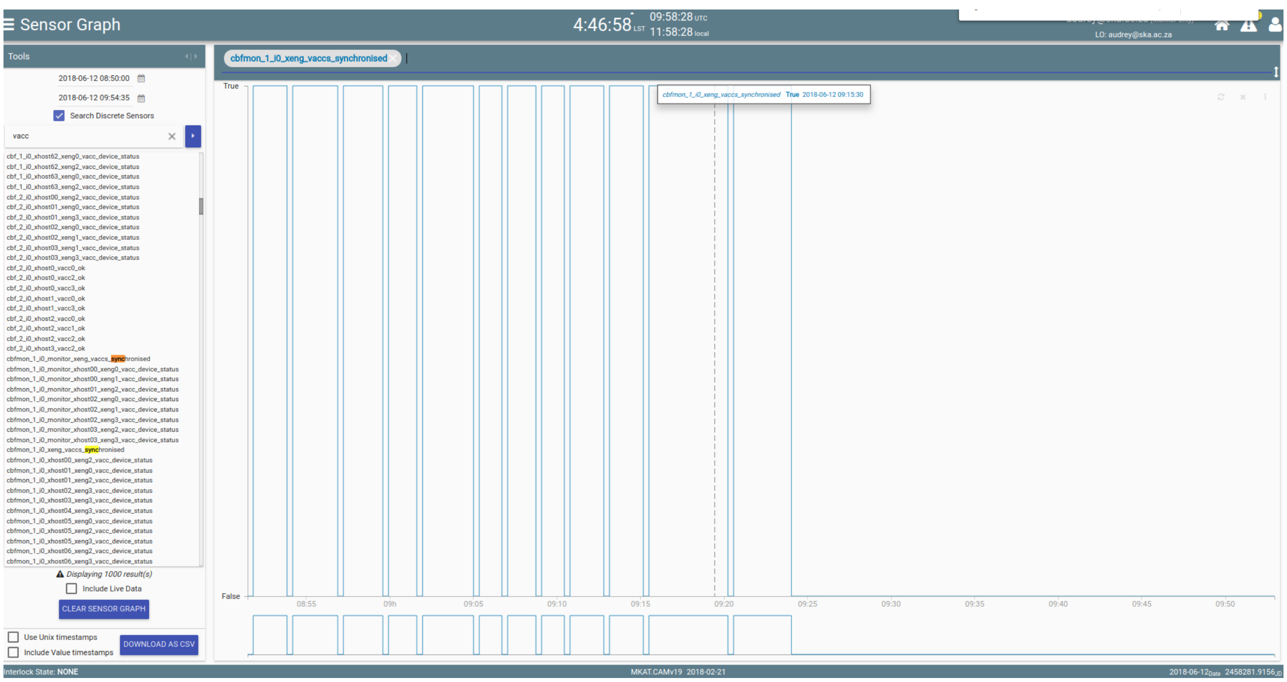


Figure 8.26: CBF vaccs sync sensor graph

8.10.10 CBF Persistent mismatched sequence numbers

- Apply procedure if only you are failing to build because of the errors shown in **Figure** 8.27:
- When cbf only reports mismatched sequence errors, halt the array instrument on cmc1 and try again. Get the array name first

  ```
  kcpcmd -s cmc1.cbf.mkat.karoo.kat.ac.za:7147 array-list
  ```

```
#log error 1528720716.452 array_1.i0 mismatched\_sequence\_number\_0x2600\_-\_expected\_0x2601
#log error 1528720716.476 array_1.i0 mismatched\_sequence\_number\_0x2601\_-\_expected\_0x2602
#log error 1528720716.499 array_1.i0 mismatched\_sequence\_number\_0x2602\_-\_expected\_0x2603
#log error 1528720716.523 array_1.i0 mismatched\_sequence\_number\_0x2603\_-\_expected\_0x2604
#log error 1528720716.546 array_1.i0 mismatched\_sequence\_number\_0x2604\_-\_expected\_0x2605
#log error 1528720716.569 array_1.i0 mismatched\_sequence\_number\_0x2605\_-\_expected\_0x2606
#log error 1528720716.593 array_1.i0 mismatched\_sequence\_number\_0x2606\_-\_expected\_0x2607
#log error 1528720716.616 array_1.i0 mismatched\_sequence\_number\_0x2607\_-\_expected\_0x2608
```

Figure 8.27: CBF mismatched sequence numbers error log

- Notice the highlighted instrument name in **Figure** 8.28

```
tiyani@tbaloyi:~$ kcpcmd -s cmc1.cbf.mkat.karoo.kat.ac.za:7147 array-list
#version-connect cbf-server CMC-M.1200.12
#version-connect katcp-device cbf-1.h v0.3.0-154-g4a427c3
#version-connect katcp-library Ckatcp-v0.2.0-204-g54eb89b
#version-connect katcp-protocol 5.0-M
#array-list array_1.i0 7413,7513 239.10.0.0+1:7148 239.10.0.2+1:7148 239.10.0.4+1:7148 239.10.0
.6+1:7148 239.10.0.8+1:7148 239.10.0.10+1:7148 239.10.0.12+1:7148 239.10.0.14+1:7148 239.10.0.1
6+1:7148 239.10.0.18+1:7148 239.10.0.20+1:7148 239.10.0.22+1:7148 239.10.0.24+1:7148 239.10.0.2
6+1:7148 239.10.0.28+1:7148 239.10.0.30+1:7148 239.10.0.32+1:7148 239.10.0.34+1:7148 239.10.0.3
6+1:7148 239.10.0.38+1:7148 239.10.0.40+1:7148 239.10.0.42+1:7148 239.10.0.44+1:7148 239.10.0.4
6+1:7148 239.10.0.48+1:7148 239.10.0.50+1:7148 239.10.0.52+1:7148 239.10.0.54+1:7148 239.10.0.5
6+1:7148 239.10.0.58+1:7148 239.10.0.60+1:7148 239.10.0.62+1:7148 239.10.0.64+1:7148 239.10.0.6
6+1:7148 239.10.0.68+1:7148 239.10.0.70+1:7148 239.10.0.72+1:7148 239.10.0.74+1:7148 239.10.0.7
6+1:7148 239.10.0.78+1:7148 239.10.0.80+1:7148 239.10.0.82+1:7148 239.10.0.84+1:7148 239.10.0.8
6+1:7148 239.10.0.88+1:7148 239.10.0.90+1:7148 239.10.0.92+1:7148 239.10.0.94+1:7148 239.10.0.9
6+1:7148 239.10.0.98+1:7148 239.10.0.100+1:7148 239.10.0.102+1:7148 239.10.0.104+1:7148 239.10.
0.106+1:7148 239.10.0.108+1:7148 239.10.0.110+1:7148 239.10.0.112+1:7148 239.10.0.114+1:7148 23
9.10.0.116+1:7148 239.10.0.118+1:7148 239.10.0.120+1:7148 239.10.0.122+1:7148 239.10.0.124+1:71
48 239.10.0.126+1:7148 239.10.0.128+1:7148 239.10.0.130+1:7148 239.10.0.132+1:7148 239.10.0.134
+1:7148 239.10.0.136+1:7148 239.10.0.138+1:7148 239.10.0.140+1:7148 239.10.0.142+1:7148 239.10.
0.144+1:7148 239.10.0.146+1:7148 239.10.0.148+1:7148 239.10.0.150+1:7148 239.10.0.152+1:7148 23
9.10.0.154+1:7148 239.10.0.156+1:7148 239.10.0.158+1:7148 239.10.0.160+1:7148 239.10.0.162+1:71
48 239.10.0.164+1:7148 239.10.0.166+1:7148 239.10.0.168+1:7148 239.10.0.170+1:7148 239.10.0.172
+1:7148 239.10.0.174+1:7148 239.10.0.176+1:7148 239.10.0.178+1:7148 239.10.0.180+1:7148 239.10.
0.182+1:7148 239.10.0.184+1:7148 239.10.0.186+1:7148 239.10.0.188+1:7148 239.10.0.190+1:7148 23
9.10.0.192+1:7148 239.10.0.194+1:7148 239.10.0.196+1:7148 239.10.0.198+1:7148 239.10.0.200+1:71
48 239.10.0.202+1:7148 239.10.0.204+1:7148 239.10.0.206+1:7148 239.10.0.208+1:7148 239.10.0.210
+1:7148 239.10.0.212+1:7148 239.10.0.214+1:7148 239.10.0.216+1:7148 239.10.0.218+1:7148 239.10.
0.220+1:7148 239.10.0.222+1:7148 239.10.0.224+1:7148 239.10.0.226+1:7148 239.10.0.228+1:7148 23
9.10.0.230+1:7148 239.10.0.232+1:7148 239.10.0.234+1:7148 239.10.0.236+1:7148 239.10.0.238+1:71
48 239.10.0.240+1:7148 239.10.0.242+1:7148 239.10.0.240+1:7148 239.10.0.242+1:7148 239.10.0.240
+1:7148 239.10.0.242+1:7148 239.10.0.240+1:7148 239.10.0.242+1:7148
!array-list ok
```

Figure 8.28: CBF array name list

- Then halt the cbf instrument

  `kcpcmd -s cmc1.cbf.mkat.karoo.kat.ac.za:7147 array-halt array_name`

- Then rebuild the subarray. If you get the same error again alert the cbf team because they may need to replace the particular SKARAB board that is causing issues.

Chapter 9

MONITORING PROCEDURES

9.1 Check CBF SKARAB Health

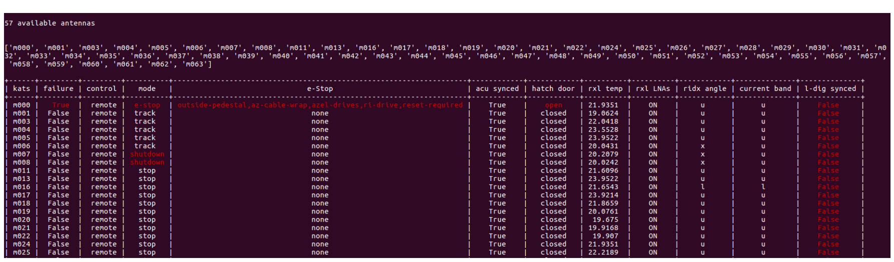
Simply follow the link to see CBF overall health
`http://cbf-nuc2.cpt.kat.ac.za/cmc1/array\_1.wide` or `http://mon-cbf.kat.ac.za/`

9.2 MeerKAT Status Sensor Summary

This too is a swift troubleshooting tool querying essential observation sensors from all active APs thus cutting down a systematic fault-finding approach.

Log onto the obs mkat machine and run the script below and the output will appear as shown in **Figure** 9.1 for L-band or **Figure** 9.2 for U-band:

```
ssh kat@obs.mkat.karoo.kat.ac.za
python usersnfs/tiyani/meerkat\_status.py --receiver-band rxl --send-status
```

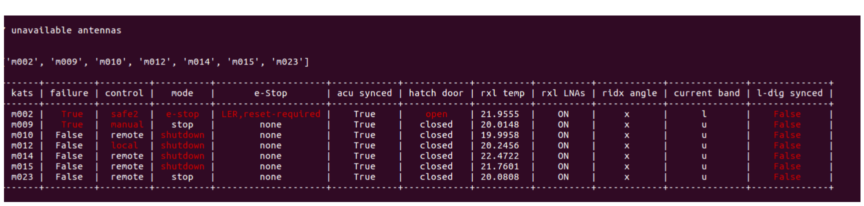


57 available antennas

['m000', 'm001', 'm003', 'm004', 'm005', 'm006', 'm007', 'm008', 'm011', 'm013', 'm016', 'm017', 'm018', 'm019', 'm020', 'm021', 'm022', 'm024', 'm025', 'm026', 'm027', 'm028', 'm029', 'm030', 'm031', 'm032', 'm033', 'm034', 'm035', 'm036', 'm037', 'm038', 'm039', 'm040', 'm041', 'm042', 'm043', 'm044', 'm045', 'm046', 'm047', 'm048', 'm049', 'm050', 'm051', 'm052', 'm053', 'm054', 'm055', 'm056', 'm057', 'm058', 'm059', 'm060', 'm061', 'm062', 'm063']

kats	failure	control	mode	e-Stop	acu synced	hatch door	rxl temp	rxl LNAs	ridx angle	current band	l-dig synced
m000	True	remote	e-stop	outside-pedestal,az-cable-wrap,azel-drives,ri-drive,reset-required	True	open	21.9351	ON	u	u	False
m001	False	remote	track	none	True	closed	19.0624	ON	u	u	False
m003	False	remote	track	none	True	closed	22.0418	ON	u	u	False
m004	False	remote	track	none	True	closed	23.5528	ON	u	u	False
m005	False	remote	track	none	True	closed	23.9522	ON	u	u	False
m006	False	remote	track	none	True	closed	20.0431	ON	x	u	False
m007	False	remote	shutdown	none	True	closed	20.2079	ON	x	u	False
m008	False	remote	shutdown	none	True	closed	20.0242	ON	x	u	False
m011	False	remote	stop	none	True	closed	21.6096	ON	u	u	False
m013	False	remote	stop	none	True	closed	23.9522	ON	u	u	False
m016	False	remote	stop	none	True	closed	21.6543	ON	l	l	False
m017	False	remote	stop	none	True	closed	23.9214	ON	u	u	False
m018	False	remote	stop	none	True	closed	21.8659	ON	u	u	False
m019	False	remote	stop	none	True	closed	20.0761	ON	u	u	False
m020	False	remote	stop	none	True	closed	19.675	ON	u	u	False
m021	False	remote	stop	none	True	closed	19.9168	ON	u	u	False
m022	False	remote	stop	none	True	closed	19.907	ON	u	u	False
m024	False	remote	stop	none	True	closed	21.9351	ON	u	u	False
m025	False	remote	stop	none	True	closed	22.2189	ON	u	u	False

Figure 9.1: Meerkat status script output for U-band



unavailable antennas

'm002', 'm009', 'm010', 'm012', 'm014', 'm015', 'm023']

kats	failure	control	mode	e-Stop	acu synced	hatch door	rxl temp	rxl LNAs	ridx angle	current band	l-dig synced
m002	True	safe2	e-stop	LER,reset-required	True	open	21.9555	ON	x	l	False
m009	True	manual	stop	none	True	closed	20.0148	ON	x	u	False
m010	False	remote	shutdown	none	True	closed	19.9958	ON	x	u	False
m012	False	local	shutdown	none	True	closed	20.2456	ON	x	u	False
m014	False	remote	shutdown	none	True	closed	22.4722	ON	x	u	False
m015	False	remote	shutdown	none	True	closed	21.7601	ON	x	u	False
m023	False	remote	stop	none	True	closed	20.0808	ON	x	u	False

Figure 9.2: Meerkat status script output for L-band

9.3 Monitor an active observation

9.3.1 Catalogues Not Matching The Observation

If and when this happens, contact the AoD. Best practice to avoid it is by running a dryrun while there are still astronomers in the office

Once the observation begins,monitor the following for "no descriptor" errors:

- **no-descriptor**-heaps-total for each band on the sdp_1 GUI sensor list.
- Go to the GUI, select sensor list and then select sdp_1.
- There are four ingest nodes where each has $\frac{1}{4}$ of the band.
- Search for "desc" and hide nominals, the no-descriptor-heaps–total should preferably remain at zero for all ingest nodes (see **Figure** 9.3) but there are times where one or two ingest nodes will increase for a very short period after the observation begins.

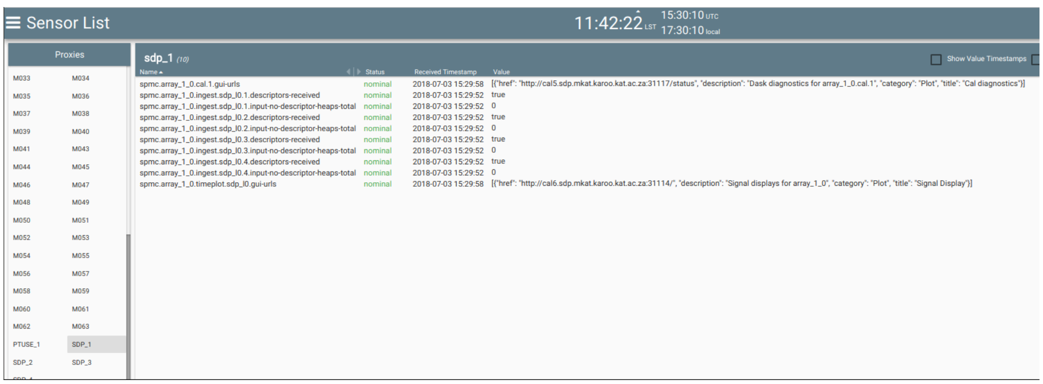


Figure 9.3: SDP sensors

- This is okay, as long as they all do not keep increasing to very high numbers.If this happens then break down the subarray, reload the Mellanox switch and rebuild the array.
- SDP Grafana GUI interface, this can be accessed via `http://mc1.sdp.mkat.karoo.kat.ac.za:3000/?orgId=1` Select the "**Overview**" option.
- There are other options which are available from the dashboard as shown in **Figure** 9.4.

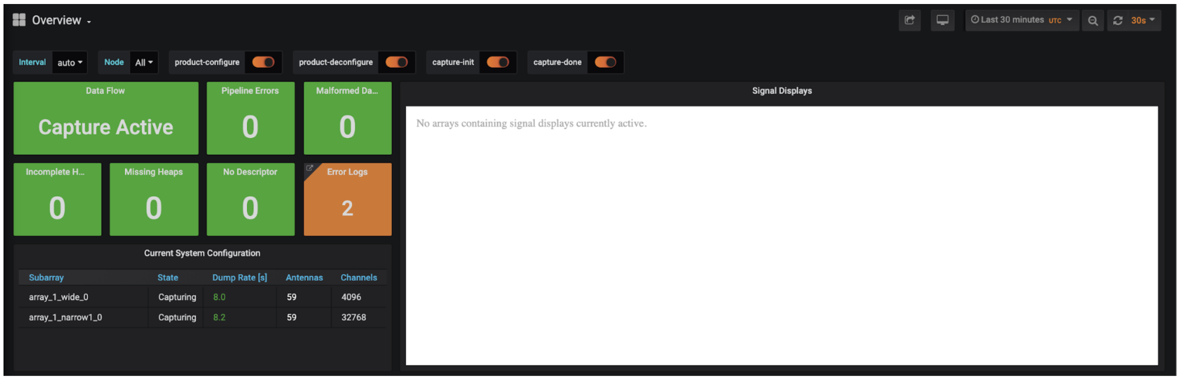


Figure 9.4: SDP Grafana dashboard

When running the observation, the dashboard will most likely show “no descriptor” errors at the beginning of the observation. No descriptor block and/or incomplete heaps block are orange, these values should return to zero within five minutes as shown in **Figure** 9.5.

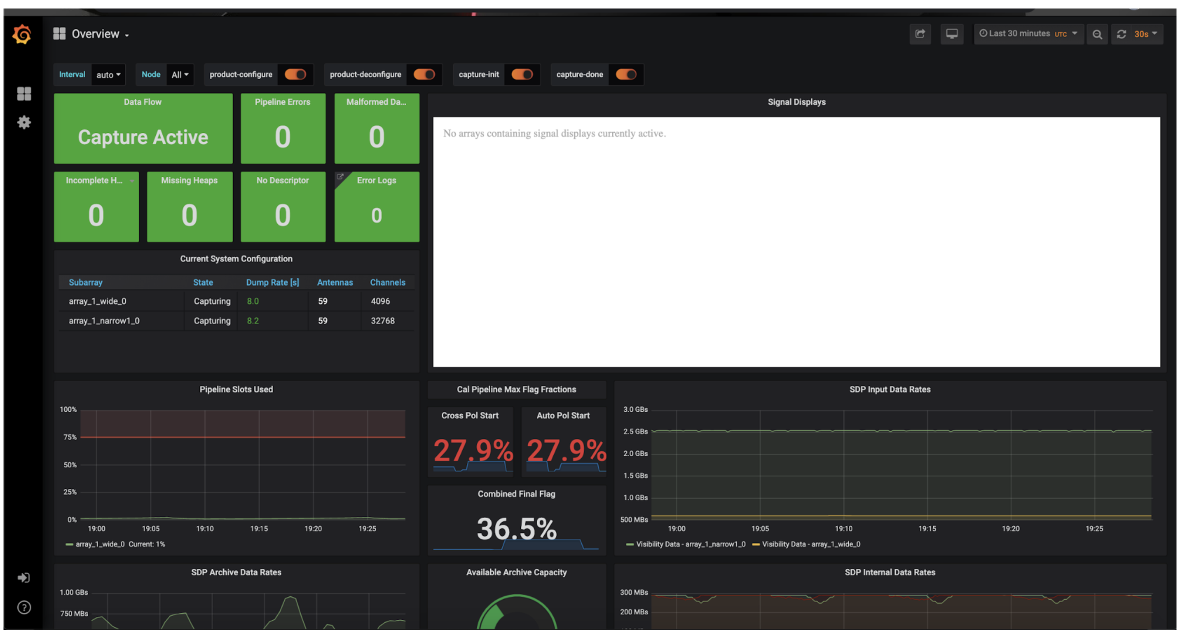


Figure 9.5: SDP Grafana dashboard without *no descriptors* errors

If the no descriptor counts keep rising, the switch needs to be reloaded

- Tail the observation progress,see **Figure** 9.6.

Figure 9.6: Observation progress log

- Load signal displays from `http://mc1.sdp.mkat.karoo.kat.ac.za:5004/` You can also click on the link shown in **Figure** 9.7, “Signal Display Dashboard ” under SDP resource.

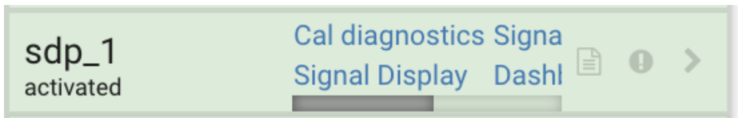


Figure 9.7: SDP signal display menu

- Type ‘load default2’ on the command line of the signal display **Figure** 9.8.

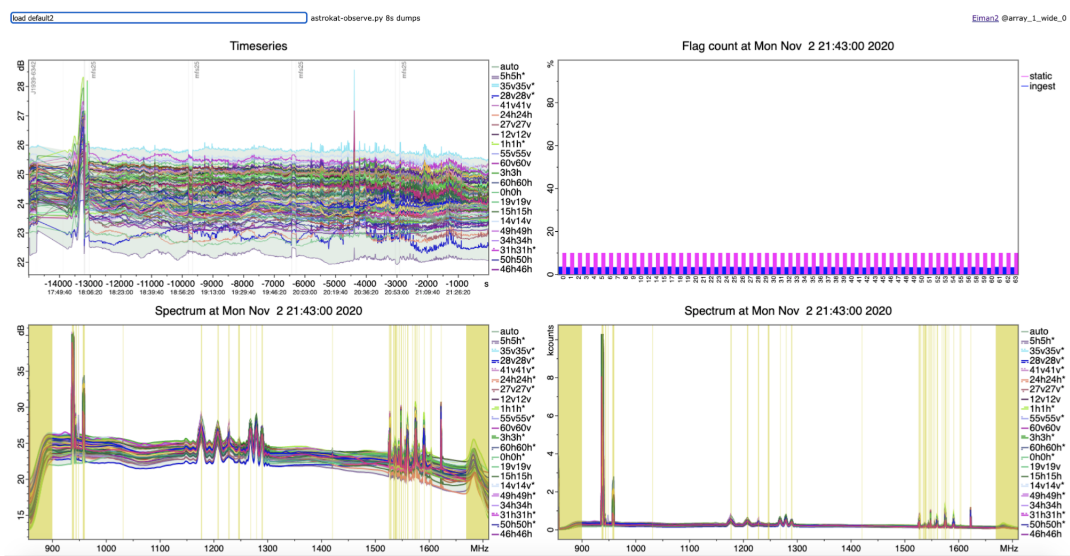


Figure 9.8: Default signal display

If on the command line you type wtabhh or wtabvv you will see displays similar to the one shown on Figure 53 and Figure 54 respectively. Here will be able to check for any faulty Receptors which are not phasing with other Receptors or may show noisy phase.

- Horizontal polarisations waterfall plot (wtabhh), this will display the waterfall in **Figure** 9.9. polarisations.

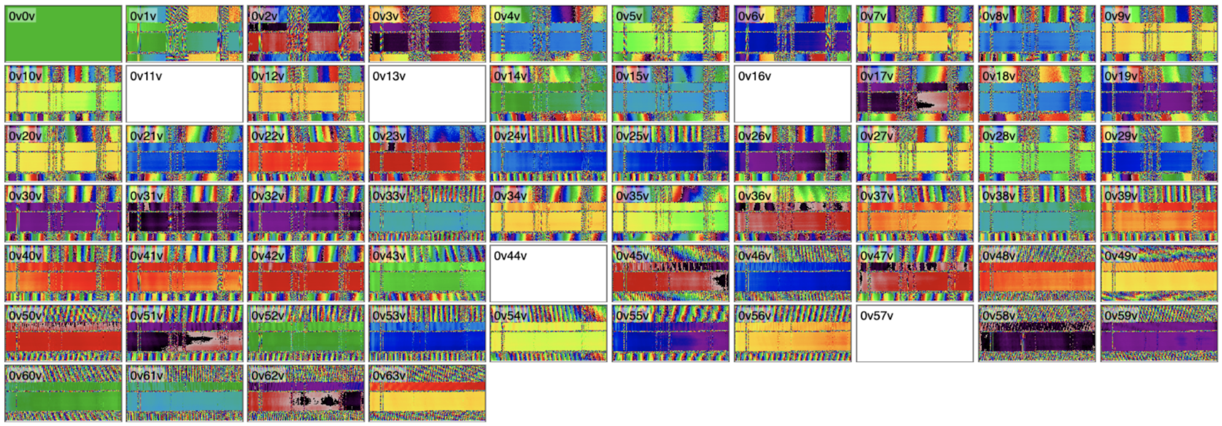


Figure 9.9: Horizontal polarisation plot (wtabhh)

- Vertical polarisation waterfall plot (wtabvv), see **Figure** 9.10

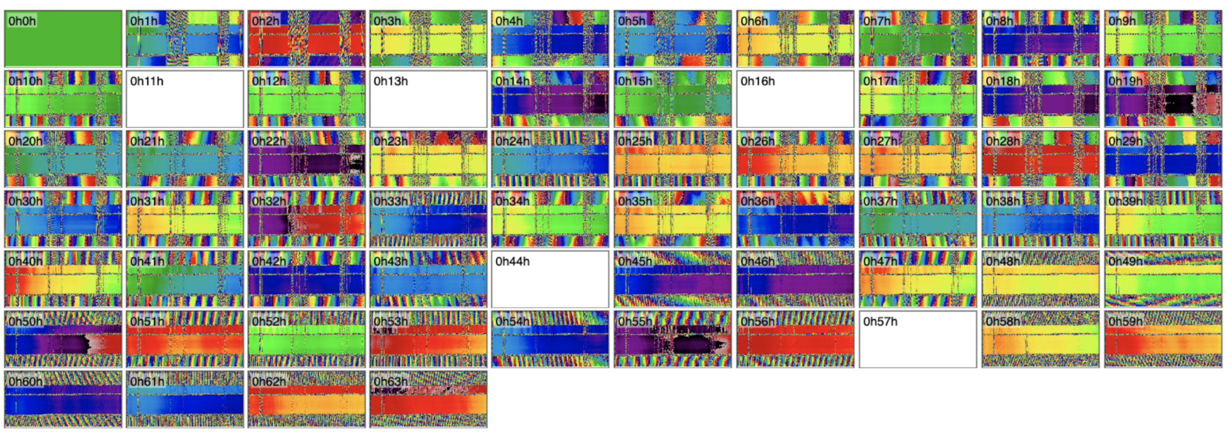


Figure 9.10: Vertical polarisation plot (wtabvv)

If for any reason, there is an antenna that is not phasing with other antennas or the phase seems noisy, then it is better to check if that antenna is moving with other antennas. You can open the receptor pointing tab and check the status of that antenna as shown in **Figure** 9.11

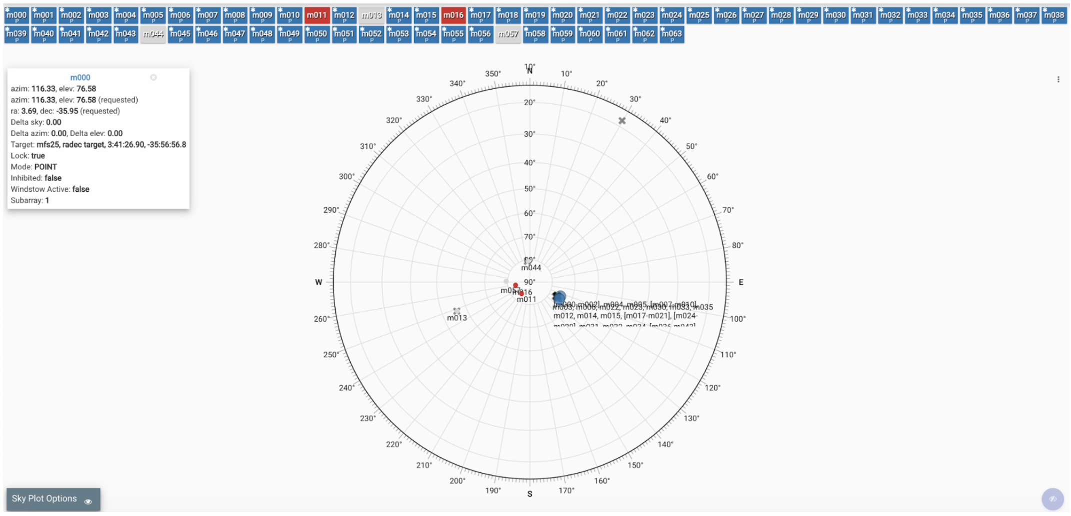


Figure 9.11: Receptor pointing plot

This will indicate to you if the antenna is locked on to the target by showing '*' on the antenna number, if it is not locked to the target there will be no '*'. It will be advisable to look at the sensor list of that antenna in questions. There may be other reasons for an antenna to have a noisy phase like a warm receiver, LNA switched off or CBF errors.

After the array is built you can select "Sensor Groups" from the CAM GUI landing page and select critical observation sensors. This will list all the critical sensors that are required to be safe and okay for the observation as shown in **Figure** 9.12.

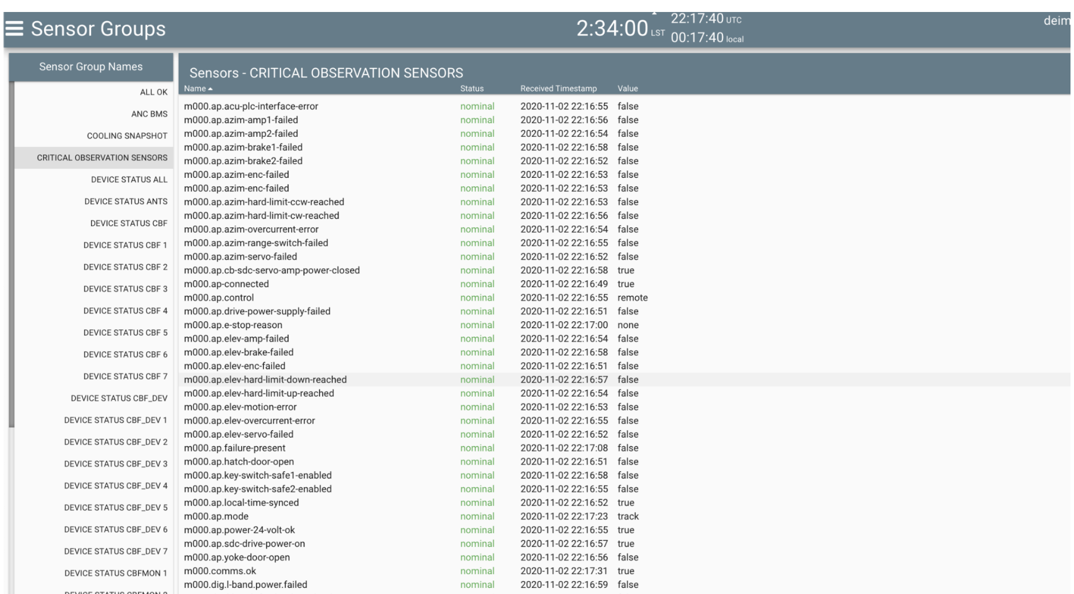


Figure 9.12: CAM GUI List of critical sensors

9.4 Monitoring SDP

In order to see the status of the SDP subsystem after the array is built and the observation is running you can start up the Science Processor dashboard (see **Figure** 9.13). This dashboard called Grafana will give you all the information about the health of the SDP system and also data flow from CBF to SDP.

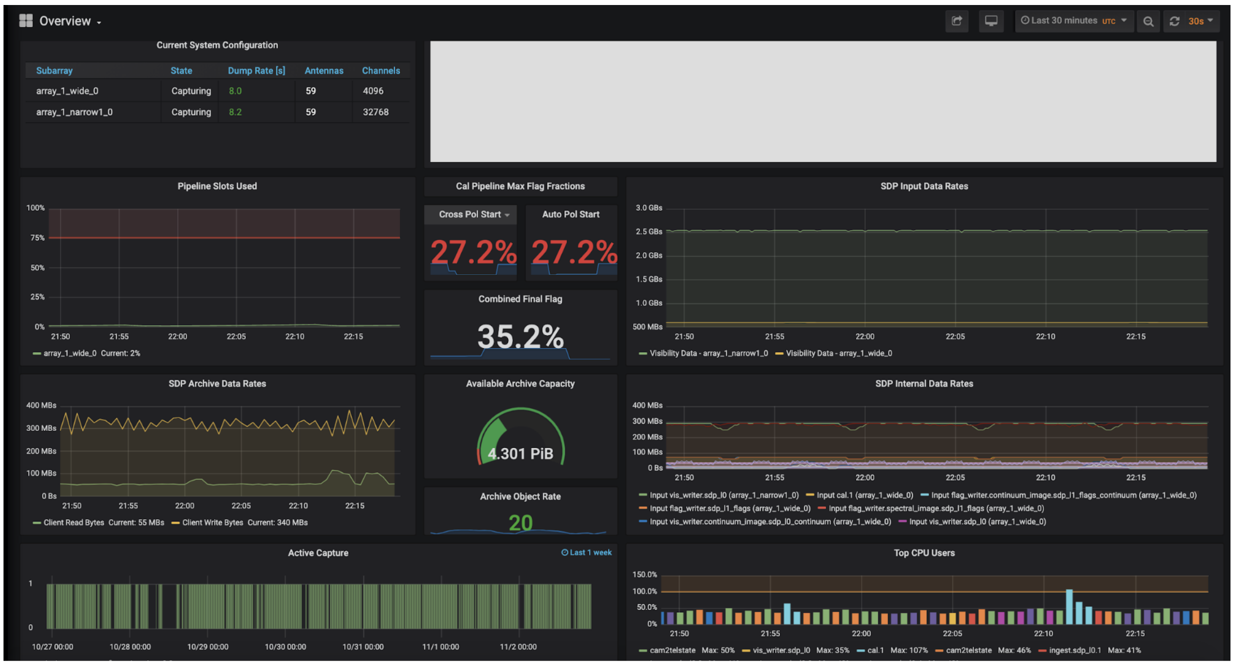


Figure 9.13: Grafana dashboard for SDP status

There are more documents that have been written which describe procedures to be followed by operators when SDP encounters errors.

9.5 Monitoring Receptors

Any faults on APs can be monitored by running the ipython session on obs.mkat.ka

```
ssh kat@obs.mkat.karoo.kat.ac.za

ipython

import katuilib
configure()

ant_inactive = sorted([ant.name for ant in kat.ants if ant.name \
in kat.katpool.sensor.resources_in_maintenance.get_value()])
ant_list = ''


for a in ant_inactive: ant_list = ant_list + a[2:] + '|'

ant_list = ant_list[:-1]

ap_sensors="(^m0(?!(%s|2(1|7).ap.ridx.hard.limit.*reached$)).*(connected|local.time|fail*|limit|(cb.*(s
```

```
kat.print_sensors(ap_sensors, strategy='period', status='warn|error|failure|unreachable')
```

This should generate the output as shown in **Figure** 9.14.

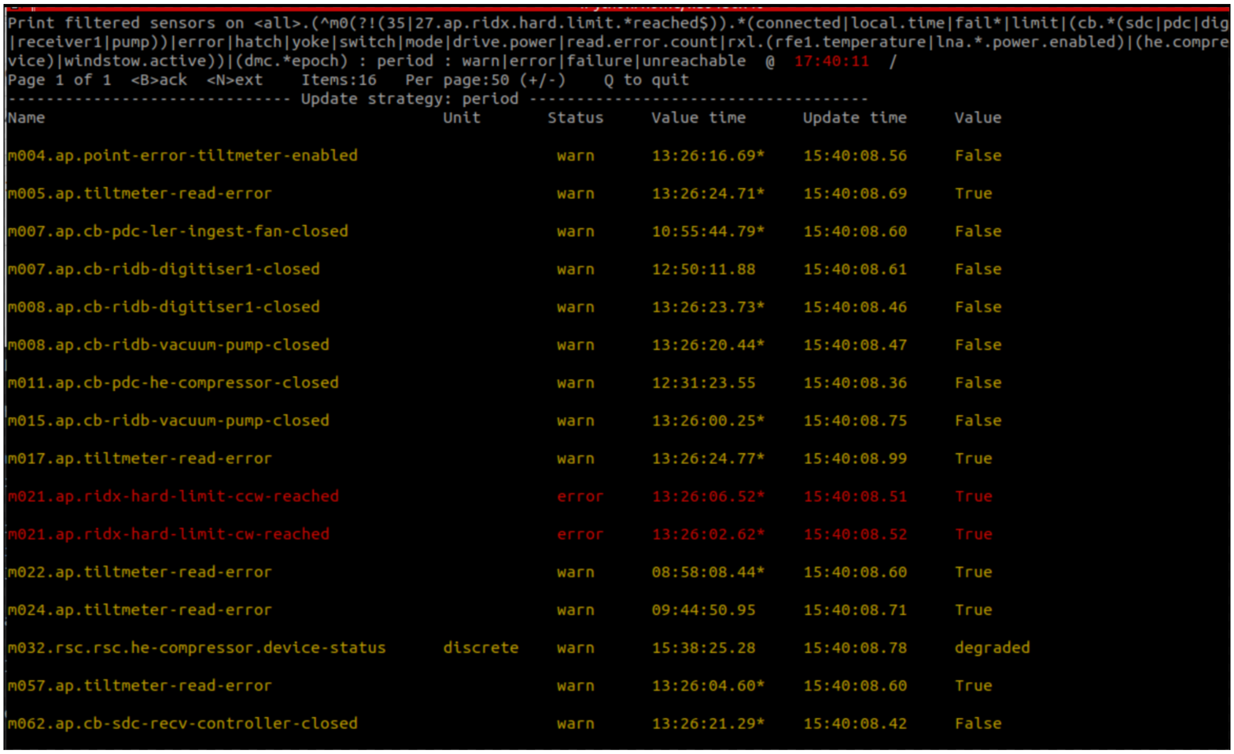


Figure 9.14: Output of antenna monitoring script

AP fault finding during observation:

- Exclude APs in Maintenance on Critical Observation Sensors

  Critical Observation Sensors is comprised of observation critical sensors. Sometimes receptors in maintenance flood the page when sensors transition to several non-nominal states, e.g. error, warn, failure, and unreachable. As this behavior is expected with maintenance, better to filter these receptors using regular expressions. The example of how to do this is shown below in **Figure** 9.15.

- Filter out nominal sensors (default setting)

  Notice m024 sensors are mainly occupying the page as it is worked on. This could potentially mask a set of active APs in the process. Moreover, m017 and m018 digitizers are inactive, would you like to constantly monitor that? Of course not.

- Excluding receptors

  For instance, `m017, m018, m024, m025` and `m063` will be excluded. The expression is: m0(?!(17—18—24—25—63)) as shown in **Figure** 9.16

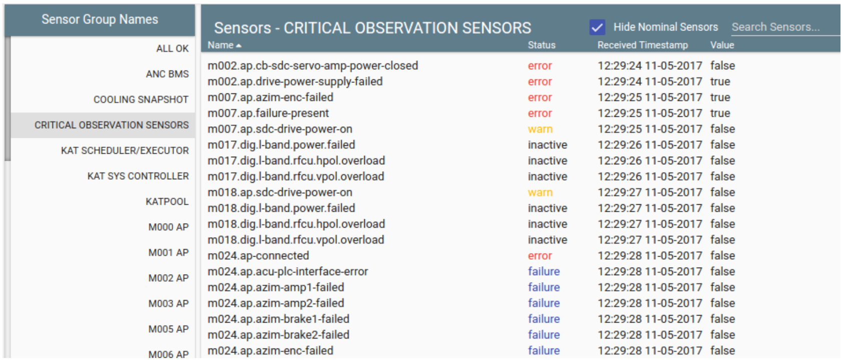


Figure 9.15: Critical observation sensors with hidden nominal sensors

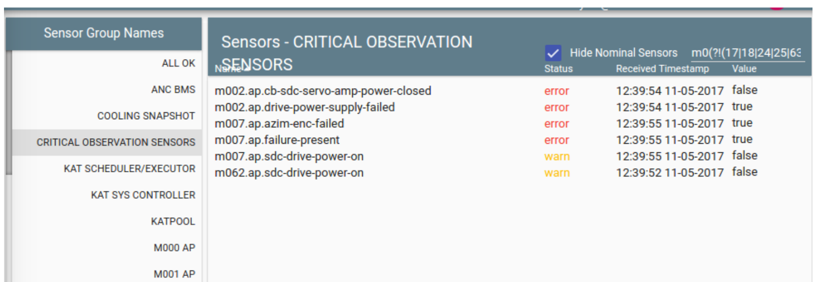


Figure 9.16: Critical sensor list filtered by AP names

9.6 Monitoring CBF

In order to monitor the status of the Correlator Beamformer (CBF) system, load the cbf mon_1 plot on the GUI to monitor CBF health as shown in **Figure** 9.17.

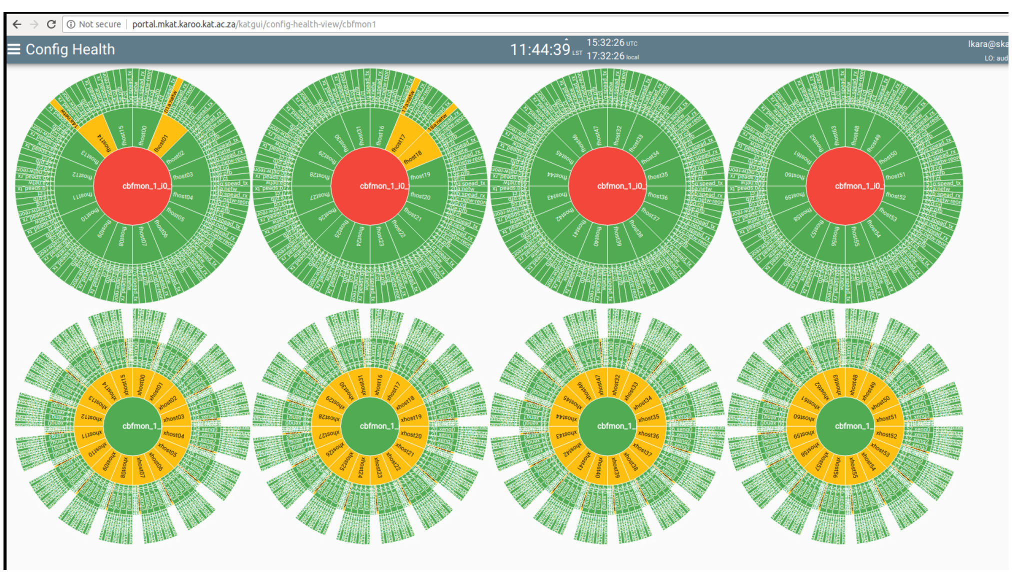


Figure 9.17: CBF monitor pie plots

This will drill down to all the sensors of the CBF and will show warnings and alarms of each Skarab board. The dashboard will give you all the f-hots and x-hosts used in the array. Note that this display only becomes live after the subarray is built. It is important to know that for a normal wide band observation with 64 antennas, there will be 128 Skarabs used. 64 are used for F-engines and another 64 are used for X-engines. But for wide band and narrow band, a total of 200 skarabs will be used for this observation.

The alternative to looking at the CBF health status as shown in **Figure** 9.17 by using the following link: `http://cbf-nuc2.cpt.kat.ac.za/cmc1/array\_1.wide`

If you want to see more CBF logs you can monitor them by running cmc-top on the obs machine.

```
cmc-top cmc1.cbf.mkat.karoo.kat.ac.za:7147
```

9.7 Monitoring Digitiser

9.7.1 Digitiser Spectrometer Overflows

This is not a critical fail - it just reports that the power into the spectrometer causes the FFT to overflow (but it still works - much like an ADC saturation) - this is not supposed to happen, but it might be that the fft_shift was decreased, causing this.

This issue is that fft_shift is set to 0 on a few digitisers (as opposed to 0xffffffff) - this will cause FFT overflows. Must be at least 7. This is TBD at this stage.

9.7.2 Digitiser Deng Overflows

When building a subarray, network problems (like the network interface crashing) can cause the digitiser to capture-start incorrectly, resulting in a deng overflow error. The f-host connected to this digitiser also a lot of times will give network reorder errors and thus appear red in the cbfmon1 plot. When you free the subarray, the error clears, but will appear again if you build a subarray again with the affected digitiser.

When this error occurs, the digitiser needs to be reconfigured: Mark the digitiser absent. Contact the digitiser team to reconfigure the problematic digitiser. Mark the digitiser ready again.

9.7.3 Digitiser Power Failed

This sensor tracks only if the system has been rebooted/lost power unexpectedly. This could mean that the digitiser has been power cycled somehow.

Since the sensor tracks lost power, it will not clear even if the digitiser has been powered again and works nominally. To clear the error: Mark the digitiser absent. Mark the digitiser ready again.

9.8 Monitoring Receiver

9.8.1 Drain current errors on the receiver

If the drain current power drops by a few mA on either the h-pol or the v-pol or both, then it will go into error. When that happens: Power cycle the LNA of that AP. Normally this will solve the problem, otherwise contact Ben Jordaan

9.9 Monitoring of Weather and Wind Sensors

9.9.1 Weather Information

There are three weather stations consisting of wind speed, wind direction, temperature and barometric pressure instruments that are installed next to the antennas. Operators are required to monitor the wind conditions as high wind conditions may cause damage to the AP. The weather information is also used for observations so it is important that operators ensure that weather instruments are working at all times. Inorder to monitor weather, click on Weather tab under sensors on the CAM GUI as shown in **Figure** 9.18

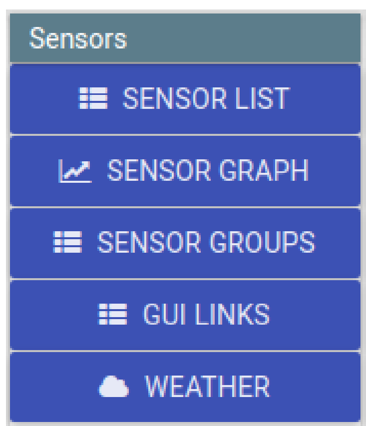


Figure 9.18: CAM GUI weather button

You will be able to view weather for the past 2 hours and up to a week interval. In the graph in **Figure** 9.19 there is a dotted line which shows the minimum and maximum wind speed threshold. If the wind speed is above minimum threshold for sustained period, the antenna will go to a stow position ¿90 deg in elevation to protect themselves from damage due to high winds. If the wind gust goes above upper limit the antennas will store immediately.

9.9.2 Weather Station Errors

ANC_Wind_Reporting_Failure The alarm is normally raised because the monitor process(mon_proxy00) is unable to sync with the ANC proxy and get the latest values. Because we do not know the latest value of the wind speed we raise an alarm: ANC_Wind_Reporting_Failure Solution to try to solve the issue: Restart the ANC proxy and the mon_proxy00 process. Should the Restart not work, call CAM Support.

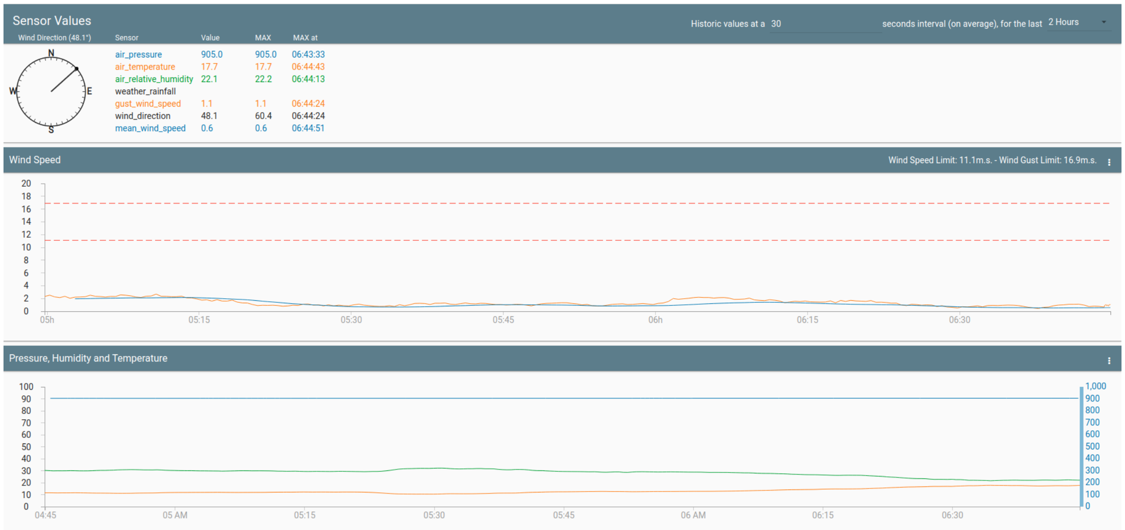


Figure 9.19: CAM GUI Weather plots

ANC_Wind_Gust The alarm is normally raised because CAM was able to read from the device that the wind speed sensor exceeds the max speed set. Solution: Nothing, except to wait for the wind speed to subside.

9.10 Monitoring BMS Sensors in KAPB

9.10.1 BMS

The BMS sensors are made to assist operators in monitoring the KAPB building that houses all the racks such as SP, APSUSE, TUSE, CBF, PTUSE, FBFUSE among others. The KAPB needs to be kept at an optimum temperature and RFI free at all times. Currently the Building Management System link to CAM is still under maintenance.
To check the BMS sensors:
In the GUI, click on the sensor list tab¿under components, click *ANC* > search "bms" or `http://portal.mkat.karoo.kat.ac.za/katgui/sensor-list?component=anc`

9.10.2 KAPB PDU Temperatures

Due to HVAC work done in the Karoo Array Processor Building (KAPB), the interface between the HVAC units and the Building Management System (BMS) is currently broken so the BMS guys are not able to monitor the temperature in the KAPB. So when you are LO please check periodically temperature sensors connected to the following Power Distribution Units (PDU). They are using this just to make sure that the correct temperature is maintained and if violated might cause temperature drifts on the Masers.

If the alarms come through please send email to Andre and Dave they will know what to do.

In order to access these KAPB PDU Temperatures you need to login to observium. The username is guest and password is guest or click the links below. A6 PDU(Hot floor level) has

an alarm but warning = 24 deg and alarm = 26 deg http://observium.kat.ac.za/graphs/to=1595401736/dev
C11 PDU(Cold below floor) has an alarm but warning = 20 deg and alarm = 22 deg http://observium.kat.ac.za/graphs/to=1595402069/device=179/type=device_temperature/from=1595315
D12 PDU has no alarm, warning = 21 deg and alarm 23 deg (this is what we must prevent, report before the temp rises to 23 deg. http://observium.kat.ac.za/graphs/to=1595399299/device=479/type=

9.11 Inspecting Telescope Network Nodes

9.11.1 Pinging Nodes With a Script

The script must be run on the ops server. It takes one of the three command-line options to the network on a subsystem-selection basis.

Start by logging onto the ops server and see the output in **Figure** 9.20

```
ssh kat@ops.kat.ac.za
cd ops\_team\_sw/utilities/mKat/
Usage helper: run
./check\_mkat\_net.sh
```

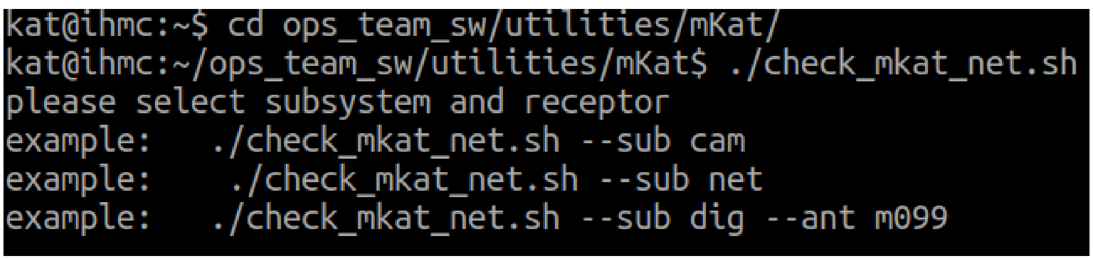


Figure 9.20: Output of _net.sh script – Spine switch status

To test all CAM nodes, run the following command and see output in **Figure** 9.21

```
./check\_mkat\_net.sh --sub cam
```

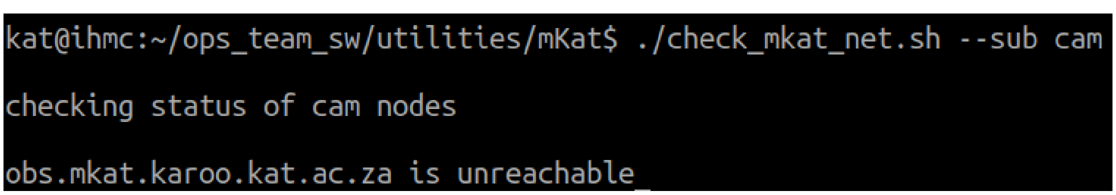
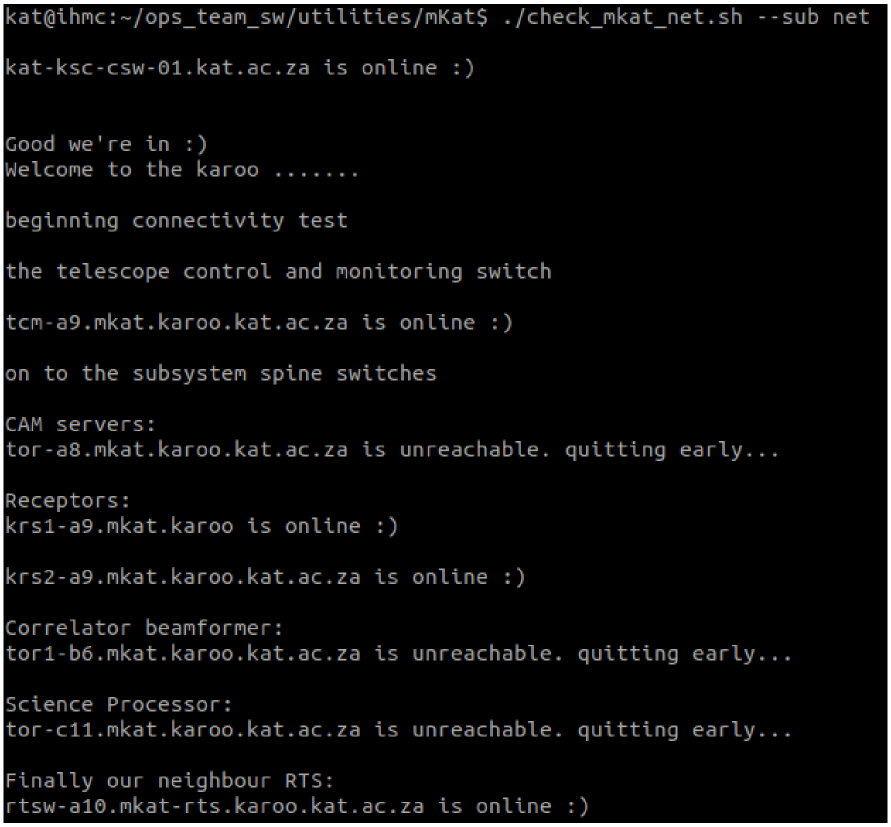


Figure 9.21: Output of _net.sh script – All CAM nodes status

To test ping the link to site and all subsystem spine switches, run the folloing command and see output in **Figure** 9.22:

```
./check\_mkat\_net.sh --sub net
```

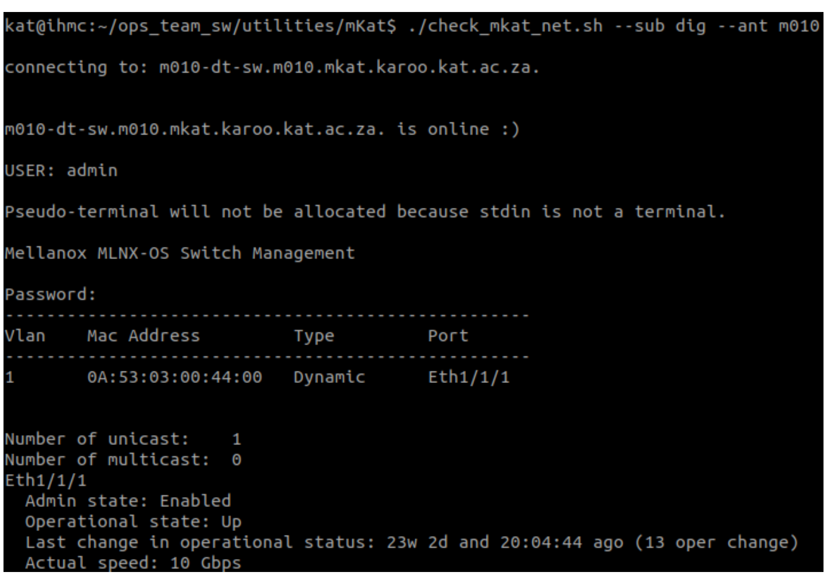


```
kat@ihmc:~/ops_team_sw/utilities/mKat$ ./check_mkat_net.sh --sub net

kat-ksc-csw-01.kat.ac.za is online :)


Good we're in :)
Welcome to the karoo .......

beginning connectivity test

the telescope control and monitoring switch

tcm-a9.mkat.karoo.kat.ac.za is online :)

on to the subsystem spine switches

CAM servers:
tor-a8.mkat.karoo.kat.ac.za is unreachable. quitting early...

Receptors:
krs1-a9.mkat.karoo is online :)

krs2-a9.mkat.karoo.kat.ac.za is online :)

Correlator beamformer:
tor1-b6.mkat.karoo.kat.ac.za is unreachable. quitting early...

Science Processor:
tor-c11.mkat.karoo.kat.ac.za is unreachable. quitting early...

Finally our neighbour RTS:
rtsw-a10.mkat-rts.karoo.kat.ac.za is online :)
```

Figure 9.22: Output of _net.sh script – Spine switch status

To check if a digitiser is online, run the folowing command and see the output in **Figure** 9.23:

```
./check\_mkat\_net.sh --sub dig --ant m0xx
```



```
kat@ihmc:~/ops_team_sw/utilities/mKat$ ./check_mkat_net.sh --sub dig --ant m010

connecting to: m010-dt-sw.m010.mkat.karoo.kat.ac.za.


m010-dt-sw.m010.mkat.karoo.kat.ac.za. is online :)

USER: admin

Pseudo-terminal will not be allocated because stdin is not a terminal.

Mellanox MLNX-OS Switch Management

Password:
---------------------------------------------------
Vlan    Mac Address         Type         Port
---------------------------------------------------
1       0A:53:03:00:44:00   Dynamic      Eth1/1/1


Number of unicast:    1
Number of multicast:  0
Eth1/1/1
  Admin state: Enabled
  Operational state: Up
  Last change in operational status: 23w 2d and 20:04:44 ago (13 oper change)
  Actual speed: 10 Gbps
```

Figure 9.23: Output of _net.sh script – Digitiser status

NOTE: L-band is connected to port: Eth1/1/1. If Eth1/1/1 is not up, power cycle the digitizer.

9.11.2 Using Observium to check the network status of nodes

An alternative to monitoring the karoo telescope network is Observium. Observium uses SNMP (Simple Network Management Protocol) to monitor and manage network device status and traffic between them. Please use a "guest" account to only monitor the network nodes.

Follow the links below to monitor node groupings of the MeerKAT Telescope network To login: `http://observium.kat.ac.za/`

Core Network: Link_to_site

Receptor switches at the KAPB: `M000-m031` and `m032-m064`

CBF Switches: Spines and Leafs

Digitizers: click the respective band on the last bulletin to load Observium. On Observium follow the routine below Select the respective receptor data switch The switch will be marked in red text if it is down Then select the "Ports" button.

```
Links to the L-band digitizer are connected to Eth1/1/x
Links to the U-band digitizer are connected to Eth2/2/x
Links to the S-band digitizer are connected to Eth3/3/x
Links to the X-band digitizer are connected to Eth4/4/x
```

U-Band L-Band S-Band X-Band

Chapter 10

Post Observation Quality Control

- Locate observation in archive - `http://kat-archive.kat.ac.za:8080/archive\_search/`
- Inspect obs report
- Rate the observation according to QA standards -`http://ops.kat.ac.za/qa/`
- Log or open a JIRA on problems picked up from the report for follow up
- Report to the AoD.

10.1 Files status via SDP MC dashboard

If an operator fails to find data files in the archive, before raising a report:

Step 1 On your preferred browser, go to `http://mc1.sdp.mkat.karoo.kat.ac.za:5004/`
This will take you to the menu as shown in **Figure** 10.1

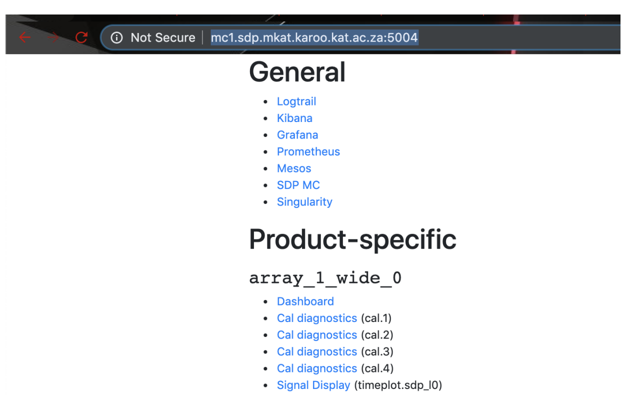
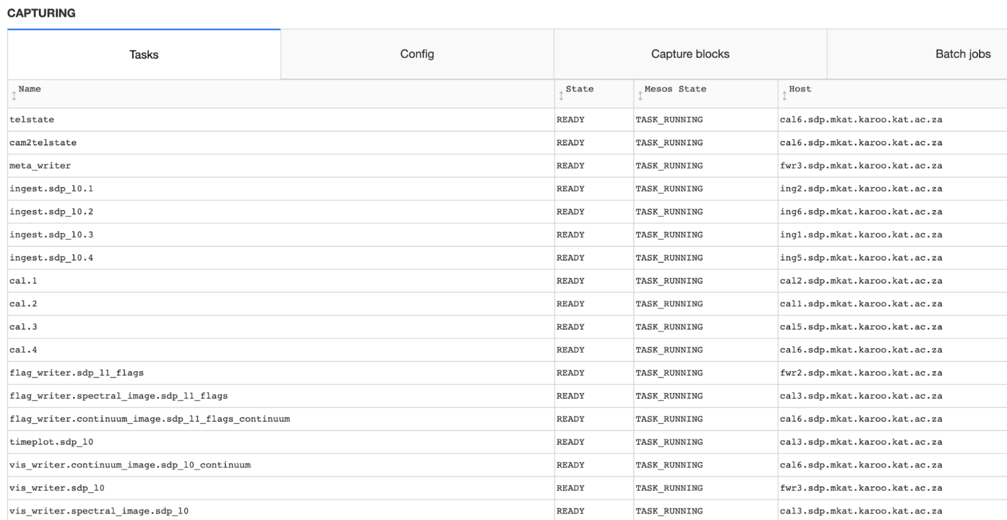


Figure 10.1: SDP main menu

Step 2 Click on DASHBOARD, this will take you to sdp dashboard as shown in **Figure** 10.2:



CAPTURING

Tasks | Config | Capture blocks | Batch jobs

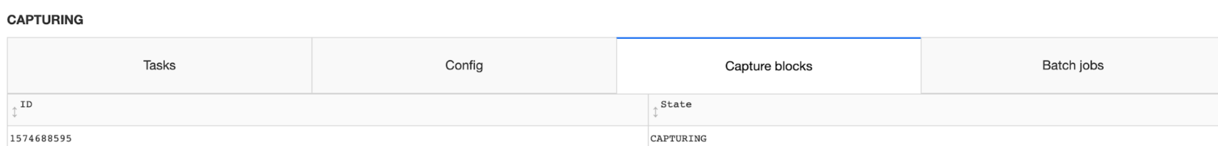
Name	State	Mesos State	Host
telstate	READY	TASK_RUNNING	cal6.sdp.mkat.karoo.kat.ac.za
cam2telstate	READY	TASK_RUNNING	cal6.sdp.mkat.karoo.kat.ac.za
meta_writer	READY	TASK_RUNNING	fwr3.sdp.mkat.karoo.kat.ac.za
ingest.sdp_l0.1	READY	TASK_RUNNING	ing2.sdp.mkat.karoo.kat.ac.za
ingest.sdp_l0.2	READY	TASK_RUNNING	ing6.sdp.mkat.karoo.kat.ac.za
ingest.sdp_l0.3	READY	TASK_RUNNING	ing1.sdp.mkat.karoo.kat.ac.za
ingest.sdp_l0.4	READY	TASK_RUNNING	ing5.sdp.mkat.karoo.kat.ac.za
cal.1	READY	TASK_RUNNING	cal2.sdp.mkat.karoo.kat.ac.za
cal.2	READY	TASK_RUNNING	cal1.sdp.mkat.karoo.kat.ac.za
cal.3	READY	TASK_RUNNING	cal5.sdp.mkat.karoo.kat.ac.za
cal.4	READY	TASK_RUNNING	cal6.sdp.mkat.karoo.kat.ac.za
flag_writer.sdp_l1_flags	READY	TASK_RUNNING	fwr2.sdp.mkat.karoo.kat.ac.za
flag_writer.spectral_image.sdp_l1_flags	READY	TASK_RUNNING	cal3.sdp.mkat.karoo.kat.ac.za
flag_writer.continuum_image.sdp_l1_flags_continuum	READY	TASK_RUNNING	cal6.sdp.mkat.karoo.kat.ac.za
timeplot.sdp_l0	READY	TASK_RUNNING	cal3.sdp.mkat.karoo.kat.ac.za
vis_writer.continuum_image.sdp_l0_continuum	READY	TASK_RUNNING	cal6.sdp.mkat.karoo.kat.ac.za
vis_writer.sdp_l0	READY	TASK_RUNNING	fwr3.sdp.mkat.karoo.kat.ac.za
vis_writer.spectral_image.sdp_l0	READY	TASK_RUNNING	cal3.sdp.mkat.karoo.kat.ac.za

Figure 10.2: SDP Dashboard

- Under MESOS STATE, you can see the status of the current observation

Step 3 Go to CAPTURE BLOCKS, this will show **Figure** 10.3.

- If there's an Observation running(CAPTURING)



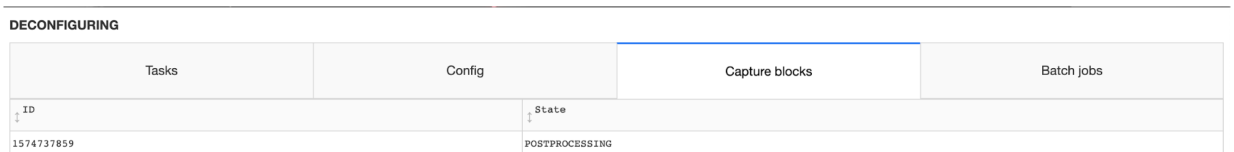
CAPTURING

Tasks | Config | Capture blocks | Batch jobs

ID	State
1574688595	CAPTURING

Figure 10.3: SDP Capturing

- If an Observation just ended(POSTPROCESSING), see **Figure** 10.2 for status.



DECONFIGURING

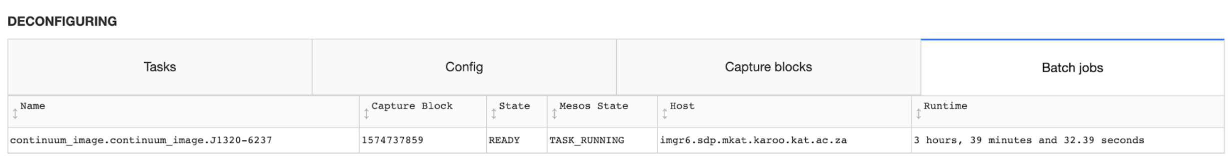
Tasks | Config | Capture blocks | Batch jobs

ID	State
1574737859	POSTPROCESSING

Figure 10.4: SDP Postprocessing

- Under Batch Jobs you can view, under Runtime, the duration of the POSTPROCESSING process as shown in **Figure** 10.5.
- Once it's done with POSTPROCESSING go to the SARAO ARCHIVE to see if the observation is now in the archive(If not, report the issue in a JIRA.

Note: Duration for POSTPROCESSING depends on the duration of the observation. (Longer observations takes longer to go through the POSTPROCESSING process)



DECONFIGURING

Tasks | Config | Capture blocks | Batch jobs

Name	Capture Block	State	Mesos State	Host	Runtime
continuum_image.continuum_image.J1320-6237	1574737859	READY	TASK_RUNNING	imgr6.sdp.mkat.karoo.kat.ac.za	3 hours, 39 minutes and 32.39 seconds

Figure 10.5: SDP batch jobs

Chapter 11

LOGGING PROCEDURES

11.1 Incidents

On the Gui under User-Logs type any currently occuring issues on the telescope, be it issues regarding building subarrays or issues with certain receptors and their performance in an observation. Use proper tags including the time-loss tag if the issue results in time being lost. The following is a list of important and relevant tags to use together with the time-loss tag:

- RECEIVER - for receiver related issues
- CBF_Resource - for issues related to cbf
- SP_Resource - for issues in SDP/SP
- AP - for AP failures and AP maintenance
- Network - in the case of network issues
- M0XX - for receptor specific issues
- Maintenance - for all maintenance work on the

11.2 Receptor Requested For Maintenance

- When a receptor goes into maintenance, a user log is automatically generated when the operator places that antenna into maintenance. It is important to keep it open-ended, i.e remove the end-time on that log and add the proper tags.
- Be sure to click the NOW button which closes the log when an open-ended log needs to be closed, as soon as that issue is resolved.
- For an issue that was closed and keeps occuring, keep updating the relevant log and update the time by pressing NOW .
- Format of the log - the tags serve as headings, what then needs to follow is the content:

11.3 What Goes Into A Log

A human-readable discussion of:

- What was being done by the operator on the telescope
- What the operator saw first and identified as symptom of an issue
- Attach screenshot, a link(to subsystem logs or jira tickets or a procedure that was followed), an email detailing the activity etc
- What the operator did in order to resolve the issue, add a link to a procedure that was followed or documented from the experience
- Close the log if the issue has been resolved by pressing the NOW button

11.4 Other Notes

Apart from writing logs, an operator needs to be a good reader of logs. You can be as good at writing logs as you are at reading them.

11.5 Reacting to anomalies

To report a fault to the relevant subsystem:

- Perform root cause analysis (any method of problem solving used to identify faults or problems)
  - Understand what the sensor is reporting
  - Time and date of fault/error
  - Description of the event or fault
  - Activity of the component in question prior to the event/fault
  - Include visual information of:
    - list showing subsystem error(s)
    - katgui alarms sensor graph
    - Grep error logs from the proxy in question
    - History or link to previous identical/similar faults
- Open a JIRA
  - Ensure the Jira has the above details from root causes analysis before logging it
  - Comment by:
    - adding the extracted logs and
    - other possible causes, the fault could link to another component or subsystem.

- Logging a fault or unusual behaviour in the user log:
  - if the resource is placed in maintenance leave the end time open until the resource is handed back healthy to operations.
- Fault correction
  - No signal displays
    - No data is being captured - investigate
  - Low output on the signal displays:
    - Check if LNA's are on
    - Check if gains are set correctly for the
    - product being used
    - Check attenuation levels
    - Check digitiser sensors
    - Check digitiser pps offset
  - High output on the signal displays
    - Check health on digitisers
    - Check using sensorgraph where the power increases
    - Check on Pointing window in GUI if there are any satellites, or find out if there could be terrestrial RFI on site
    - Check digitiser pps offset
  - No fringes
    - Check if delay tracking is on
    - Check delay model, normal value for the cable delay should be approx 5800...
    - See the observer sensor on receptors
    - Verify that the delay calibration script has been run after the subarray activation
  - AP Failure
    - An AP failure can be seen in the progress of the observation or CAM flagging on flagcount.
    - Check the Sensor List of the AP in the GUI - hide all nominal sensors.
    - If not clear from Sensor List what the issue is, go to `http://ops-k1.ops.karoo.kat.ac.za:8081/localfile/Auto\_Reports/Servo` For logs go to:

      ```
      ssh kat@portal.mkat.karoo.kat.ac.za
      Cd /var/kat/log
      ```

    - Draw a sensorgraph of the fault and open JIRA. This procedure includes when AP is not locking on target.
  - AP not locking on target
    Risks

  * AP stuck and not moving - not collecting requested data.
  * Antenna number reduction - reduced number of baselines, compromised U/V coverage

Procedure

  * In Receptor Pointing, verify that the ap is not locking (Lock: True or False
  * Record the time from the Gui for the purpose of Log searching in step 4
  * Check the AP Sensor list,hide nominal and warning status, then see which AP sensors are in error, probably there is none except ap device health
  * Reset AP failures in obs machine and do the following:
  * Check the AP mode in the sensor list, if ap mode is in shutdown, set the ap mode to stop otherwise it will not move.

    ```
    ssh kat@obs.mkat.karoo.kat.ac.za, ipython, import katuilib,
    configure\_cam('camcam','all')
    cam.m0xx.req.ap\_reset\_failures()
    cam.m0xx.req.mode('STOP')
    ```

    The AP should start receiving commands and moving

    ```
    ssh kat@portal.mkat.karoo.kat.ac.za
    cd /var/kat/log
    less filename(e.g kat.m007.2020-08-31.log)
    ```

    OR use `grep` commands to search for a particular string within a file. Using the timestamp in step 2, look at what sensors went into failure,error or degraded state. (Name your JIRA as per sensor name in error)
  * Draw a sensorgraph to verify if it was a real error and for how long it has been so
  * Open Jira and populate it with all your findings, assign it to Audrey to moderate.
  * If all fails - check to see antenna proxy logs to see if it is updating or not - if not restart the AP proxy from the GUI - UNDER PROCESS CONTROL.

11.6 Creating a Jira

- Record the following (Also see section Above "Reacting to anomalies":
  - Date and time of when the error occurred.
  - The "Description" of the observation that was running.
  - The "Proposal ID" of the observation that was running.
  - Attached the link to the progress report of that current observation.
  - A Screenshot of the signal display recording the behaviour of the antenna in error.
    * Make use of the following filters on the Signal display
      - Load default2 (plotting the spectrum of only the antenna in question)
      - Load default2 (plotting the antenna in question against other antennas that is in nominal state) Wtabhh and wtabvv

        * A screenshot of the sensor list showing the sensor that is in error.
            · Draw a sensorgraph or the error, confirming the time when the antenna went into error and when the error cleared. (Helps to Identify error behaviour and occurence)
            · Attach any other related information recorded in screenshots to the ticket.

- On the Jira System, Click create, then:
    - Select "Project":
        * Operations (OPS)
    - Select "Issue Type":
        * Select "Failure", to report any failure.
        * Select "Task", if you're assigning a Job for someone to complete/do.
    - Summary:
        * Write one sentence describing the error/irregular behaviour and include the number of the antenna. ns (OPS)
    - Components:
        * Select the component in question by searching through some of the available tags. Select:
            · L-band RX, for l-band receiver related issues.
            · U-band RX, for u-band receiver related issues.
            · AP, for AP related issues.
            · etc.
    - Labels:
        * Almost the same as above, but in this case you select the labels of the component/s in question
            · l-band , for l-band related issues.
            · UHF, for u-band related issues.
            · etc.
    - Description:
        * Re-iterate the start time of the error, the description of the observation, the error that occured and on which antenna the error occurred.
        * Describe and list all actions taken by you, the Lead Operator, in an attempt to clear the error.
    - Assignee
        * Assign the Jira to the contact person of the subsystem in question, If not sure, assign the ticket to Audrey or Clifford.
    - After Creating the Jira make sure to add all related subsystems as watchers. And also tag your fellow operators in the ticket.

11.7 Who to assign the JIRA to?

Table 11.1: List of JIRA asignees

Component	**Description**	**Component lead**
CBF	Issues that affect moving SKARABS, data loss, spectrum,etc. Anything relating to the correlator or beamformer.	Alec Rust
KDRA	Karoo Array Processor Building.	Andre Walker
CAM	Control and monitoring Software and GUI issues.	CAM Support
Operations	Issues relating to Operations.	Clifford Gumede
S-Band Dig	This refers to the S-band packetiser which is like a digitiser	Dave Horn
USE	All User Supplied Equipment (TUSE, FB-FUSE, APSUSE and BLUSE) related issues.	Dave Horn
U-Band Dig	Issues relating to the UHF digitiser	Henno Kriel
L-Band Dig	Relate to L-band digitiser errors,mainly hardware sensors like PPS	Henno Kriel
U-Band Rx	Issues relating to the UHF band receiver	Ben Jordaan
L-Band Rx	All issues relating to the L-band receiver system, including RSC and bandpass issues.	Ben Jordaan
S-Band Rx	Refers to the s-band receiver. The receiver is supplied by MPI and is managed by Pieter Kotze and Ben Jordaan.	Ben Jordaan
TFR	Time and Frequency Reference System (TFS)	Johan Burger
Digitiser	Refers to general digitiser issues and is not band specific. Issue could be with global sync or any software which Marc can assist with.	Marc
Servo System	Refers to all servo system parts, like servo amplifiers, servo motors, encoders, azimuth switch and all motor failures	Mike Chalmers
AP	Refers to antenna positioner(AP) physical structure or mechanical components, like, structural damage, pointing issues, etc.	Matthys Maree
S-band System	Refers to faults which occurred during an s-band observation, including CBF, CAM, SDP, TFR and signal chain for s-band issues.	Pieter Kotze
SDP	Refers to issues relating to components of the Science Data Process (SDP) subsystem such as Signal Displays, Grafana, SARAO Archive, and sometimes script failures.	Rosly Renil

Chapter 12

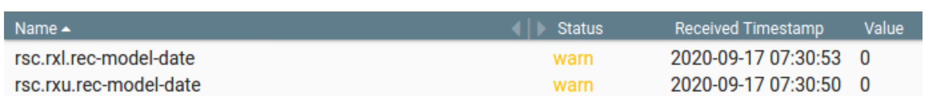
Receptor Acceptance And Verification

For donwloads of:

- pointing models, contact OOD,
- delay models, contact AOD
- receiver models, see procedure below.

12.1 Receiver Models Updates

Step 1 Whenever a receiver is replaced on a receptor, check the ' *rsc.rx?.rec-model-date*' sensor (where *?* is the band) of that receptor as shown in **Figure** 12.1. Compare the value you found with the value given in the table found in this document. If there is a difference, the receiver models need to be updated and thus proceed to step 2 If the values are the same, no action is required.



Name	Status	Received Timestamp	Value
rsc.rxl.rec-model-date	warn	2020-09-17 07:30:53	0
rsc.rxu.rec-model-date	warn	2020-09-17 07:30:50	0

Figure 12.1: Receiver model date sensors

Step 2 A script is in place to download and push the models to GitHub. Do the following:

```
$ ssh user@mkat-thn-linux.mkat.karoo.kat.ac.za
```

- Here "user" refers to your FreeIPA username (the one you use to log in to sites like EduVPN and Mattermost). When it asks you for a password, use your FreeIPA password.

```
$ cd ../msteyn/download\_rx\_models
$ ./run\_rx\_models_download.sh --ant m0xx --band (l or u)
```

- This will output an instruction with receiver serial number (see **Figure** 12.2). Copy or add the serial number as the second argument.

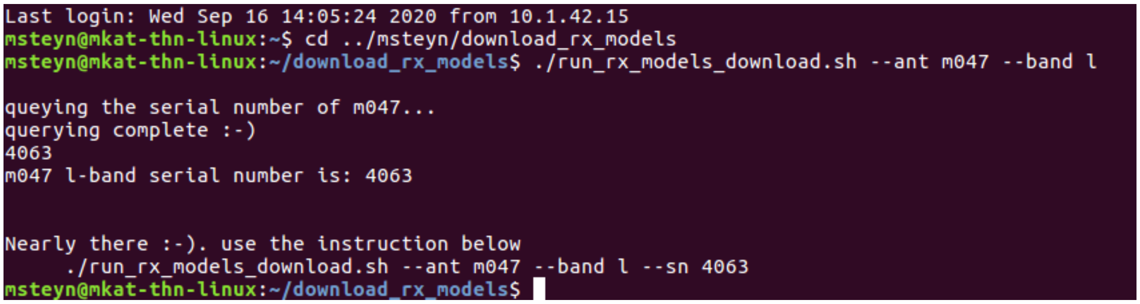


Figure 12.2: Receiver model downloads

```
$ ./run_rx_models_download.sh --ant m0xx --band (l or u) --sn xxxx
```

- At some point the script will prompt you to enter your FreeIPA username (it's the one you use to log in to sites like EduVPN and Mattermost).
- When prompted enter "rsc" as the password to download the models to your local directory. See **Figure** 12.3 for the output.

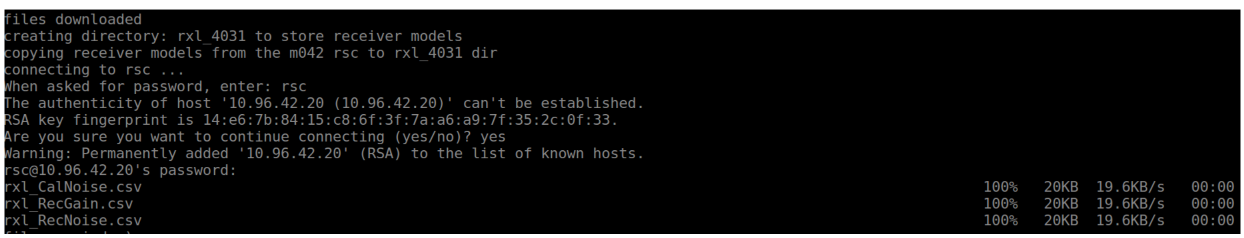


Figure 12.3: Receiver model files downloaded

- Later you will be asked a few times to enter your GitHUb credentials (username and password) as shown in **Figure** 12.4.

Figure 12.4: Receiver model password

- When the script gives the message as shown in **Figure** 12.5, then it was successful in uploading files to GitHub and you can proceed to step 3.

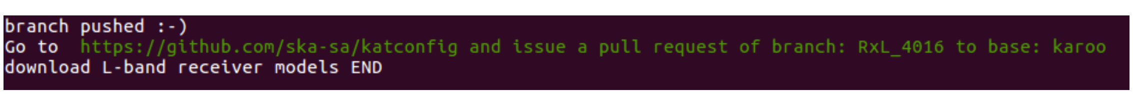


Figure 12.5: Receiver model git push succesful

Step 3 Issue a pull request on GitHub to merge the branch created in step 2 above to the karoo branch as follows:

- Go to `https://github.com/ska-sa/katconfig` where you will land at the page in **Figure** 12.6:

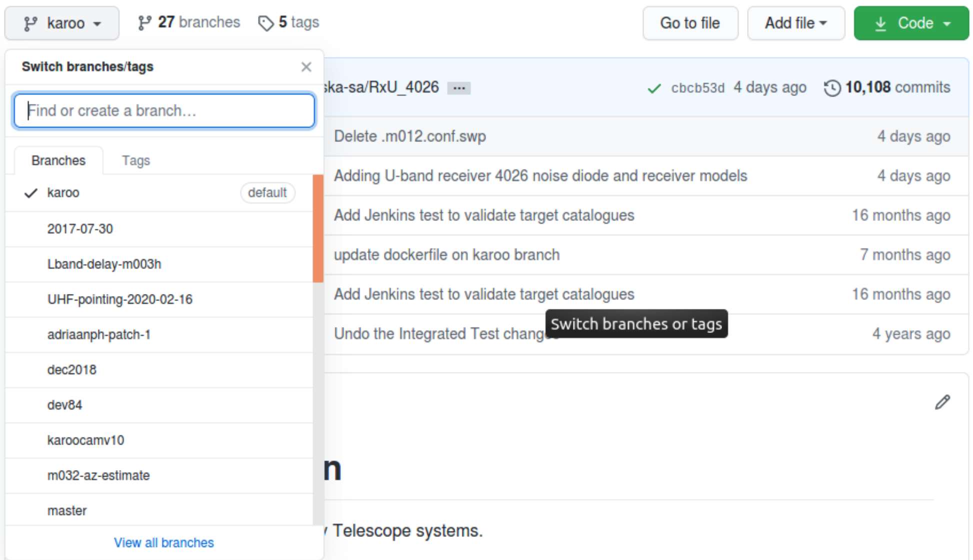


Figure 12.6: Receiver model github compare and pull request

- Click on "Compare & pull request".
- Make sure "base: karoo" and "compare: Rx(L or U)_xxxx" is selected. Then write a comment stating what you are doing, why (give Jira number if applicable) and make the request out to Pieter Kotze. See **Figure** 12.7 for an example:

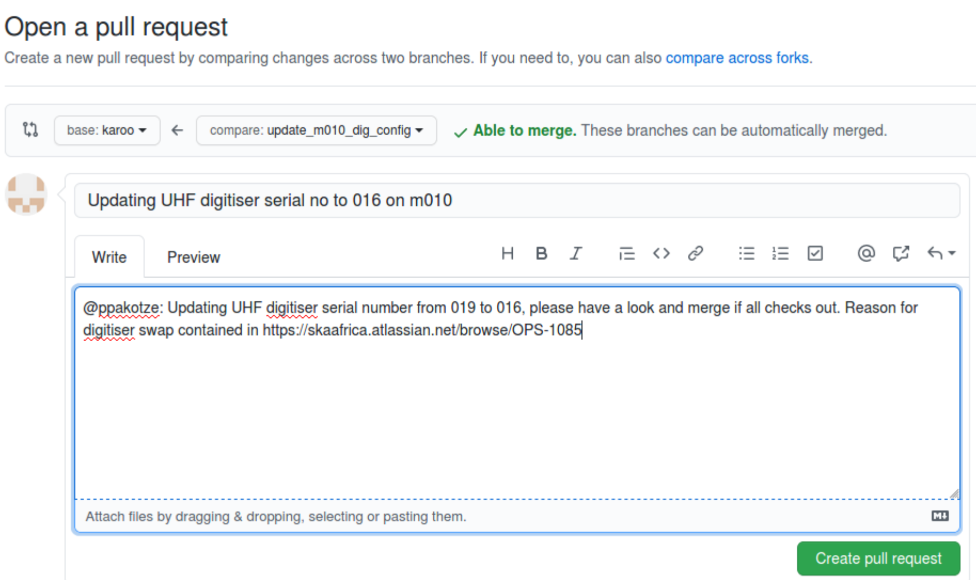


Figure 12.7: Receiver model github pull request

- At the right hand side at the top next to "Reviewers", click on the gear icon. It will drop down a list of reviewers to choose from. Type "ppakotze" in the search field (see **Figure** 12.8) to find Pieter K and then click on Pieter K to request him to approve your pull request (It will show a check mark next to his name, and then under "Reviewers" you will see a yellow dot next to his name).

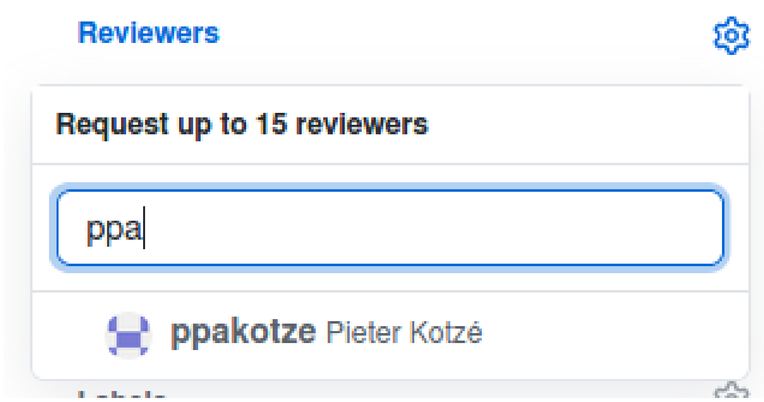


Figure 12.8: Github add pull request reviewers

- Then click on "Create pull request". Note: After Pieter K approves the pull request, he usually also merges the pull request, but sometimes you have to do the merging yourself. You do this by going to the pull request after it has been approved, and then click on "Merge pull request" at the bottom and then "Confirm merge".
- Proceed to step 4.

Step 4 Update the table found in this document with the latest value of the "rsc.rx?.rec-model-date" sensor (where ? indicates the band) for the antenna and band whose receiver models were updated.

12.2 Checklist after new receiver installation

1. Check for Ben's handover from the specific jira
2. Check *rsc.rxl.startup-state* from sensor list - it should be cold-operational
3. Check *rsc.rxl.rfe1.temperature* from sensor list which should around 19 K
4. Check receiver models and compare with those in the katconfig as per procedure above.
5. Run reset attenuations - if power is out of range then proceed to step 6 otherwise proceed to step 7.
6. Run refine attenuations
7. Run delay calibrations
8. If the results from delay cal are too high or too low create a jira to Marcel Gouws

12.3 Receiver Vacuum Pump Oil Lubrication

CAUTION: *Prevention or little oil available in key receiver areas that need to be lubricated. The lubrication is necessary over a period of 2 weeks. Include as many antennas as possible since CBF and SDP are not required. If the antennas are not lowered down this could cause serious problems.*

- Wait for the flight departure on site. Needs to be above 16 deg C ambient as well (NB for winter)
- Build a subarray with many free antennas as possible
- Leave out the receiver band, cbf and sdp
- Lower the antennas 18 degrees elevation. Use the receptor flight stow script
- If antennas go into stow or someone moves the antenna to higher elev while the script is busy running, please notify Ben Jordaan about the event.
- If the script is interrupted before Vacuum pumps are enabled, notify Ben urgently.
- Report to Ben which antennas did not start / stop the script.
- If the Vacuum pump lubrication script does not work / is interrupted on a certain antenna notify Ben.
- Run the schedule below to lubricate the receivers

```
configure_obs()
obs.sb.new(owner='MeerKAT')
obs.sb.type = katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.antenna_spec = 'available'
obs.sb.controlled_resources_spec = ""
obs.sb.description = 'MeerKAT: Lubricate Vacuum Pumps'
obs.sb.instruction_set = "run-obs-script /home/kat/katsdpscripts/utility/lubricate_vacuum_pump.py -n 'o
obs.sb.notes= "projectID=RTS "
obs.sb.to_defined()
obs.sb.to_approved()
```

12.4 S-band Operations

Note: When there is a schedule for S-band testing, Pieter Kotze would usually attend and either do the S-band testing himself, or ask you to build the S-band array and run observations with clear instructions either in the calendar or directly to you on IRC.

The following are most important sensors to check in order to know which antennas are usable for S-band building:

- The *rsc_rxs_temvac_temp15K* sensor as shown in **Figure** 12.9 can be used to determine if S-band receiver is cold. If the above sensor is under 30 K, then the receiver is usable.

Figure 12.9: S-band receiver temperature sensor

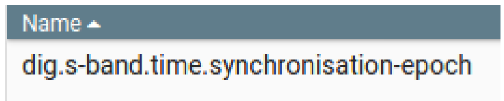


Figure 12.10: S-band time synchronisation epoch sensor

- The *dig.s-band.time.synchronisation-epoch* sensor as shown in **Figure** 12.10 can be used to check if the S-band digitiser is synced correctly.

If the sync epoch is not correct (is not the same number as the others), then an absent/ready cycle is needed on the digitiser for it to receive the right epoch.

Note that the conventional commands for marking digitisers absent/ready do not work yet for S-band digitisers, hence you need to use the following commands on obs machine:

- Make sure the “ *dig-selected-band*” sensor is not currently on ‘s’.
- To mark the digitiser absent, use

  ```
  kcpcmd -s 10.103.254.2:72xx digitiser-status s absent
  ```

- To mark the digitiser ready, use:

  ```
  kcpcmd -s 10.103.254.2:72xx digitiser-status s ready
  ```

  In the steps above `xx` indicates the antenna number. Once the above mentioned sensors are all correct, the antennas should be usable for S-band observations.

- Note: A quick way to check the above mentioned sensors is with these commands in an ipython session:

  ```
  cam.print_sensors('temp15k')
  cam.print_sensors('s.band.time.synchronisation.epoch')
  ```

12.5 How to build an array

NOTE: *Only dashes will be displayed next to “NGS” on the GUI when building in S-band. It’s normal for now and does not mean that the S-band digitisers are not synced. As mentioned before, if the epoch of the digitisers are correct, they are synced and ready.*

Usually you build an array with:

- Product = c875M4k (Band = s)
  - Note: 4k and 32k builds in S-band are normally stable, but only 4A 1k S-band builds work (other 1k S-band builds still have CBF issues).

- Centre frequency = 3062.5MHz (S4)
  - Note that SDP only supports these center frequencies for S-band: 2843.75 MHz (S3) and 3062.5 MHz (S4) and must be selected manually on the CAM GUI (does not get chosen by default).
- Dump rate = 1s
- Add cbf_1 (CMC1) and sdp_1 resources if you are on subarray 1. Only use CMC2 if instructed to do so.
- Usable Antennas: Check the OPS Catalyst notice board for the latest on which antennas can be used in S-band as shown in **Figure** 12.11.

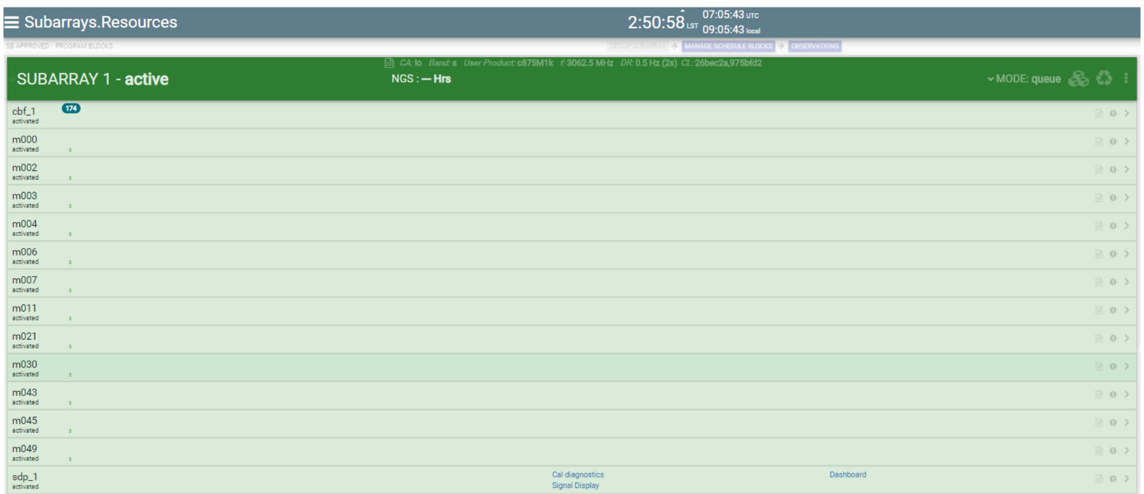


Figure 12.11: Example of a S-band array

12.6 S-band Observations

12.6.1 Delay Calibration

Then run a delaycal:

- Example of a good delaycal: `http://10.97.1.13:8081/tailtask/20200929-0021/progress`
  - Note: `M056` vpol has an unstable signal power (visible on the timeseries), however you still include AP in S-band observations (**don't mark it faulty**).

If you see strangeness on antennas not AP failure related (saturation, weird power levels, noisiness, large delay offsets, no delay solutions etc), keep them in (don't mark them faulty) for Single dish RTS observations (phase stability, drift scans, tipping curve etc.) as RTS still wants their data. Can only exclude the APs for observations such as S-band holography.

12.6.2 Example of schedule blocks

Description:

```
obs.sb.new(owner='RTS')
obs.sb.type = katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.description = 'RTS: S-band Drift Scan of Hyd A'
obs.sb.antenna_spec = 'available'
obs.sb.controlled_resources_spec="cbf, sdp"
obs.sb.proposal_id = 'COMM-RTS'
obs.sb.instruction_set= "run-obs-script /home/kat/usersnfs/benjamin/drift_scan_2.py 'hyda' --drift-dura
obs.sb.notes= "reductionid='N/A' , projectID=RTS "
obs.sb.to_defined()
obs.sb.to_approved()
```

12.6.3 Phase stability

While you have S-band built phase stability could also be run on Hydra A. Just remember it will need a delay_Cal.

```
obs.sb.new(owner='RTS')
obs.sb.type = katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.description = 'RTS: 3.1.2.1 S-band Interferometric Phase Stability hyda '
obs.sb.antenna_spec = 'available'
obs.sb.controlled_resources_spec="cbf, sdp"
obs.sb.proposal_id = 'COMM-RTS'
obs.sb.instruction_set= "run-obs-script /home/kat/katsdpscripts/RTS/2.9-RFI/track.py 'hyda' --repeat -t
obs.sb.to_defined()
obs.sb.to_approved()
```

12.6.4 Tipping curve

Additionally a tipping curve could be useful. Same array configuration, can be run anytime during the day or night and takes about 1.5 hours.

```
obs.sb.new(owner='RTS')
obs.sb.type = katuilib.ScheduleBlockTypes.OBSERVATION
obs.sb.description = 'RTS: 2.1 S-band Tipping_Curve'
obs.sb.antenna_spec = 'available'
obs.sb.controlled_resources_spec="cbf, sdp"
obs.sb.proposal_id = 'COMM-RTS'
obs.sb.instruction_set= "run-obs-script /home/kat/katsdpscripts/RTS/2.1-Tipping_Curve/obs_tipping_curve
"
obs.sb.notes= "reductionid='2.1' , projectID=RTS "
obs.sb.to_defined()
obs.sb.to_approved()
```

Chapter 13

Accessing Site and Apps

13.1 Hosts and Servers

- Obs machine - where schedule blocks are created, system commands are run, system health is gathered via sensor filtering etc,

○ *kat@obs.mkat.karoo.kat.ac.za*

- Portal machine - where all historical and current system logs are obtained at *kat@portal.mkat.karoo.kat.ac.za*. See GUI links in **Table** 13.1.

- CBF CMC1 and CMC2 logs (run from the obs machine)

```
cmc-top cmc1.cbf.mkat.karoo.kat.ac.za:7147
cmc-top cmc2.cbf.mkat.karoo.kat.ac.za:7147
```

- DMC Telnet - see all digitiser logs

```
telnet 10.103.254.2 7147
```

Table 13.1: Links to Meerkat, Kat7 and RTS portals

Instrument	**KatGUI**	**m0xx Digitiser**	**m0xx Digitiser Switch**
MeerKAT	`http://portal.mkat.karoo.kat.ac.za`	`http://10.97.1.14`	*10.103.254.2:72xx* `ssh admin@10.96.xx.40`
RTS	`http://portal.mkat-rts.karoo.kat.ac.za` `http://10.97.8.3`		
KAT7	`http://portal.kat7.karoo.kat.ac.za` `http://192.168.193.11`		

13.2 Utilities and Apps

- KatGUI video feed
  `http://portal.mkat.karoo.kat.ac.za/katgui/video`
- Servo System debugging tool
  `http://ops-k1.ops.karoo.kat.ac.za:8081/localfile/Auto_Reports/Servo`
- Receivers debugging tool
  `http://ops-k1.ops.karoo.kat.ac.za:8081/localfile/Auto_Reports/Receiver/Daily`
- Scheduler
  `http://mkat-scheduling.kat.ac.za/`
- Archive
  `http://kat-archive.kat.ac.za:8080/archive_search/`
- Source Elevation Notebook
  `http://kat-imager.kat.ac.za:8888/notebooks/AR1_reduction_results/Utilities/Source%20Elevation.ipynb`
- Site Activities
  `https://docs.google.com/spreadsheets/d/1QNhA2IQ7aMkVMqF2MDGLyVMAWyN9cK5RrtsI-AysknI edit#gid=191831361`

Chapter 14

Reporting in Operations Meetings

- **Meeting Document**
  - After the Monday operations meeting, an OOD or any operator assigned by the OOD should open new operations meeting minutes for the week.
  - Ensure that you copy the template from the 2021 Operations meetings and verify that the new copy is saved in the new folder.
  - The name of the document is the current Monday date to the following Monday date as in **Figure** 14.1.

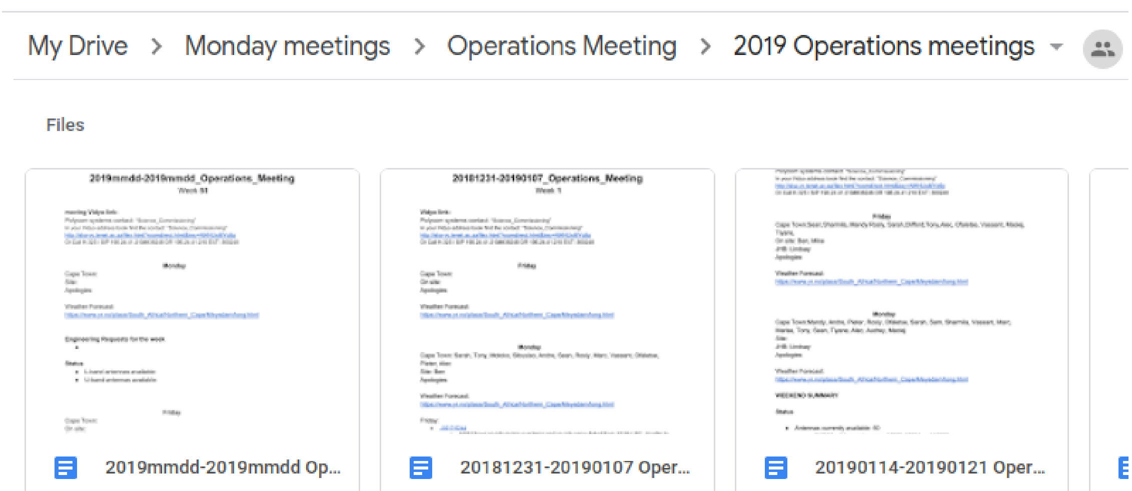


Figure 14.1: Operations meeting document google folder

- **Status**
  - How many operational antennas are available currently in both bands L and UHF?
  - How many antennas are currently in maintenance, today? Reason? When are they returning?
  - Link each issue to a JIRA

  - Always remember to report trends/frequencies of similar issues.
  - Divide issues in each section by the days which they occured.

- **CAM/SP/CBF**
  - What are the latest changes/deployments/hot-fixes done this week?
  - What are some of the issues experienced from CAM on the system, are they resolved? If not what is the current status of the problem?

- **Digitisers/ Receivers and APs**
  - Are there latest software/firmware updates done?
  - Any other recent work done on each AP,dig or RSC?
  - What are the issues on each APs? What did they affect? How long were they in erroneous state for? Has the JIRA been attended to?
  - What is the resolution if there is one? Is there a procedure from this JIRA? You can also choose to include some of this information under the observation session..... Where they are most relevant as to when they happened.

- **Integration and Testing**
  - Ensure that what you are reporting to have happened in the week actually has happened and perhaps on the Thursday or Friday morning ask the people responsible for that task for feedback for this meeting, they can fill it in themselves in the minutes if you do not understand the details.

Notice Board If there are any issues that you will like other operators to be aware of please add to the electronic notice board at `https://docs.google.com/document/d/1YvHi7U63WUbkbUGOQRW\_PJcZZlRzpOYb6mnv2CZlkSM/edit`

Appendix A

Checklist

Starting your shift:

Did you read the handover on OPS-CATALYST?
Did you read the Notice board?
Have you reserved antennas for site maintenance?
Will the sync epoch time suffice for the current schedule block?
Have you opened mattermost communications tool.
Did you run AP Meerkat status script to check availability of AP?

Subarray Configurations:

Is the band, data product,dump rate,cbf,sdp and USE as per calendar request?
Have you run reset attenuation after adding antennas from maintenance?
Have all antennas calibrated for delays? if not, repeat and/or mark faulty ones?
Have you verified if phase-up must be run from calendar request?

Observation:

Is the dryrun valid, if not consult AOD
Are signal displays plotting?
Do you follow the progress report of the observation
Are observation files closed without errors
Is the observation file in the archive?
Have you checked the cal report and conducted the QA?
Have you opened JIRAs for encountered system problems?
Have you created/closed any timeloss logs on the telescope?
Have you updated the observation document and Operations Minutes?
Have you updated the Ops catalyst with the number of antennas available?

Appendix B

Connecting to EDU-VPN

B.1 Procedure for Linux

- Go to `https://kat-cpt-vpn.kat.ac.za`
- Log in using your credentials provided to you.
- Go to the" *Configurations*' link.
- Click on 'Continue...', type a name in the 'Name' field and then click 'Create and Download'. This will create a .ovpn config file with the name as provided in the 'Name' field and prompt you to download the file.
- When prompted, save the file somewhere on your drive (the 'Home' directory is a good place as the terminal usually opens up with the Home directory as the default directory)
- Assuming you have OpenVPN installed, open a terminal session and type:

  ```
  sudo openvpn --config 'filePath/fileName.ovpn'
  ```

  where 'filePath' is the directory in which the config file is saved and 'fileName' is the name of the config file downloaded. For example if the name of the config file is 'SiteVPN' and it was saved in the home directory of a computer whose user is called 'admin', then the command will be as follows:

  ```
  sudo openvpn --config 'home/admin/SiteVPN.ovpn'
  ```

  Once command is executed and the last line is:

  ```
  Initialization Sequence Completed
  ```

  then you are connected.
- If you do not have Openvpn installed, then follow the steps below:
  - After downloading the .ovpn config file, go to VPN settings as shown in **Figure** B.1.

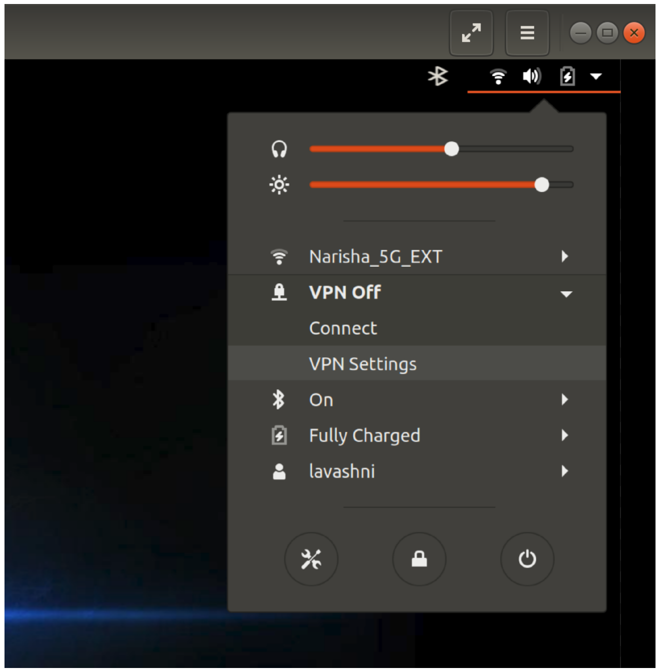


Figure B.1: Linux VPN settings

- Click on the 'plus' sign to add VPN as shown in **Figure** B.2

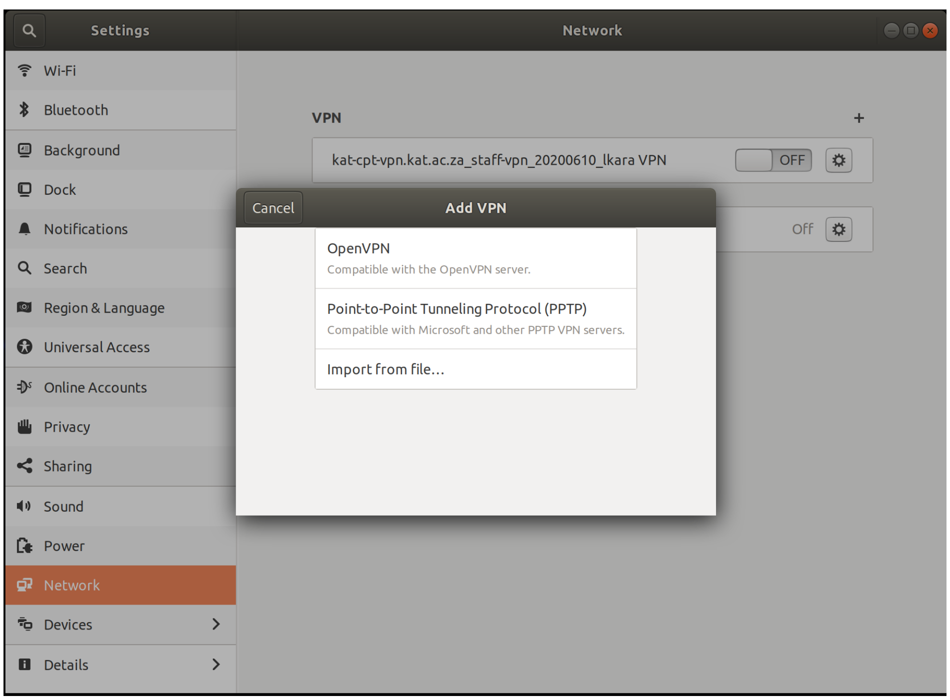


Figure B.2: Linus add vpn dialog box

- Select 'Import from file...' (see as shown in **Figure** B.3), you will be directed to your 'Home' folders so that you can choose the correct file to import

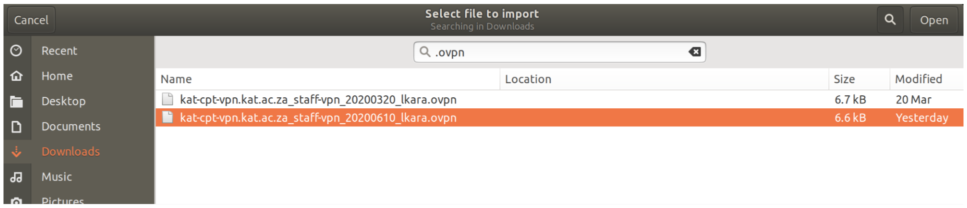


Figure B.3: CAM GUI approved SB

- After clicking on the correct .ovpn config file, the window as shown in **Figure** B.4 should appear

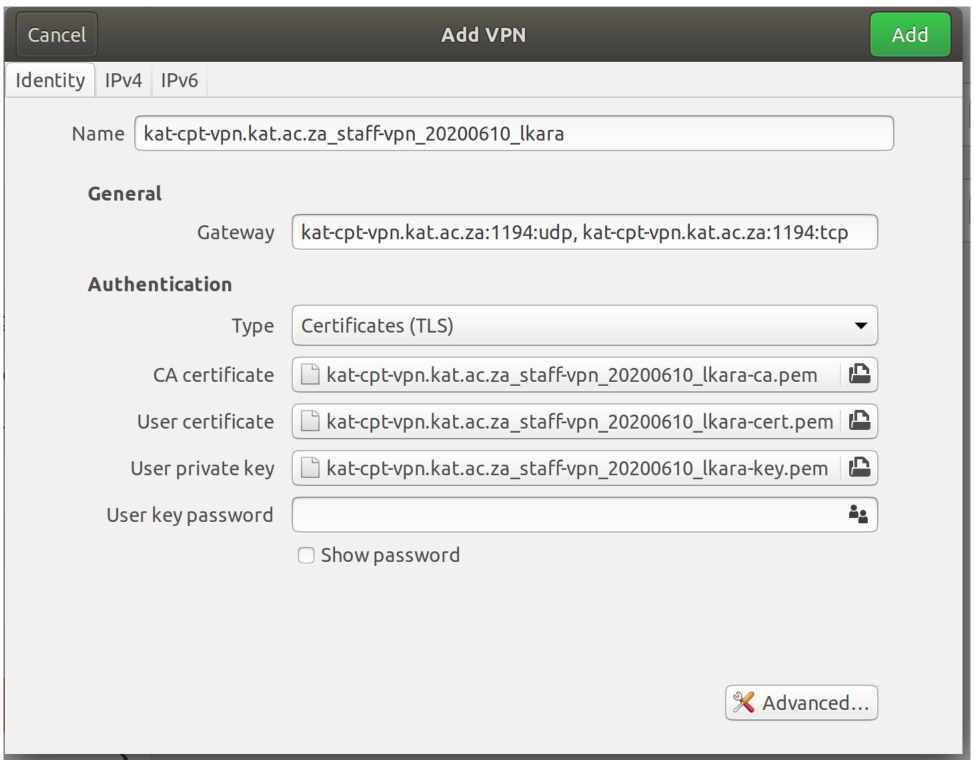


Figure B.4: CAM GUI approved SB

- All of the information from the .ovpn config file has automatically been filled in, all you need to do is click 'Add'.
- You will see the name of the latest .ovpn config file installed as shown in **Figure** B.5.

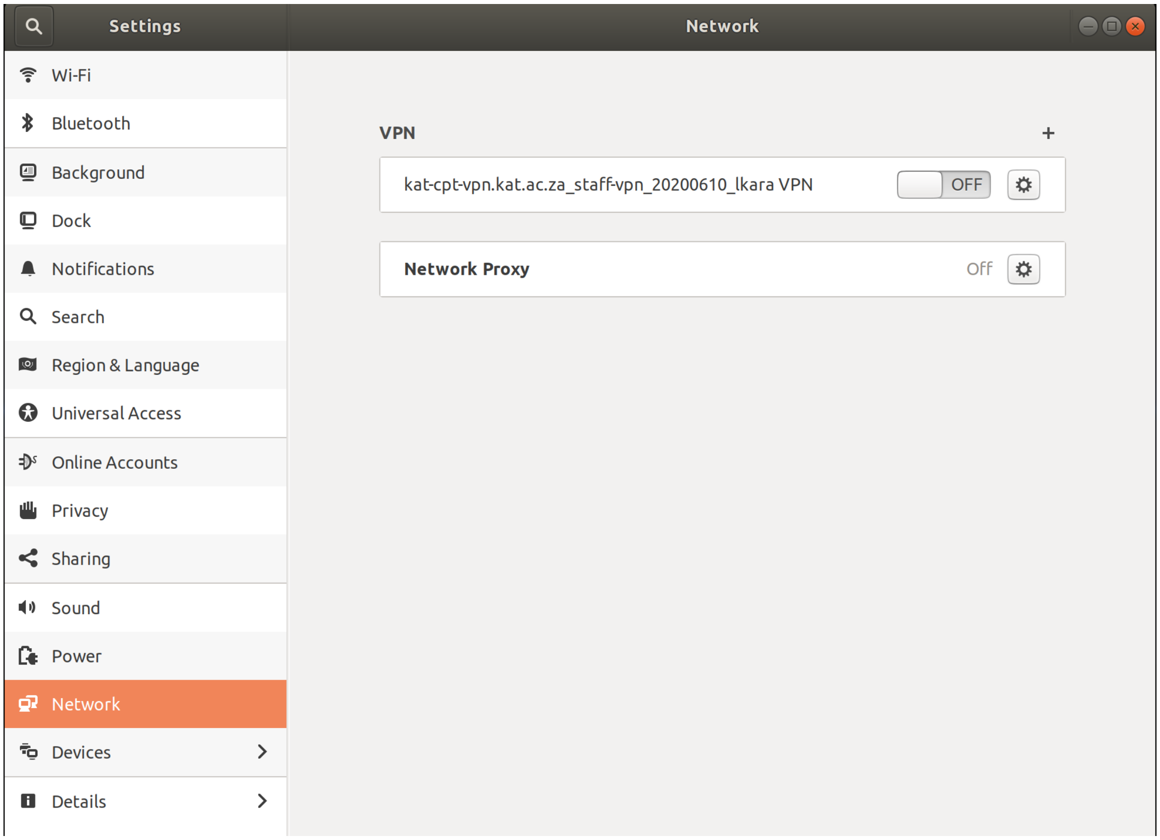


Figure B.5: Name of latest EduVPN configurtion file

- You can now connect to VPN (see **Figure** B.1)

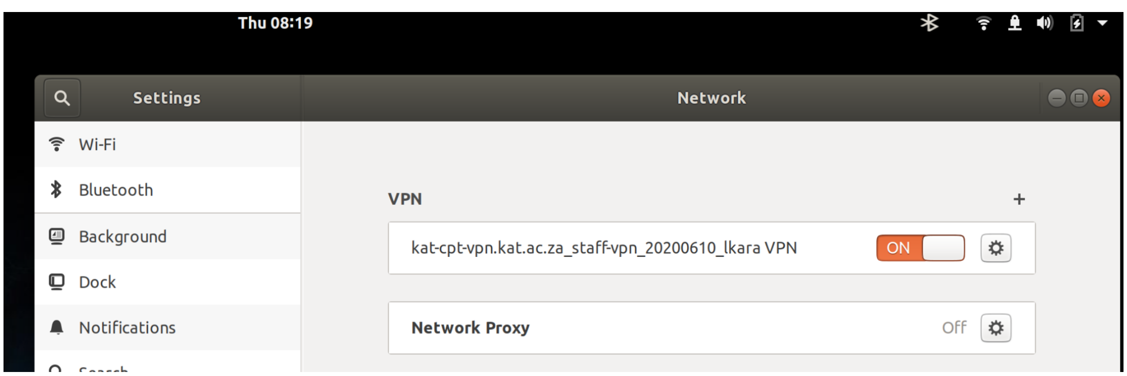


Figure B.6: VPN connected

- Don't forget to delete the old configuration every time that you download a new one on the EduVPN site as shown in **Figure** B.15

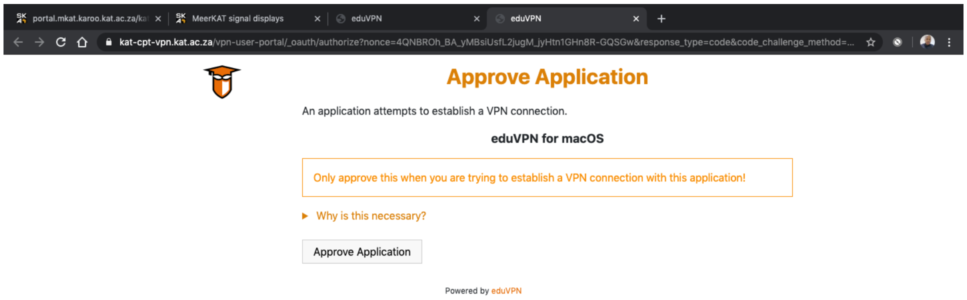


Figure B.7: Linux EduVPN approve dialog box

B.2 Procedure for MacOS

Download the EduVPN macOS Software from `https://kat-cpt-vpn.kat.ac.za/vpn-user-portal/home` and the home should appear as in **Figure** B.8.

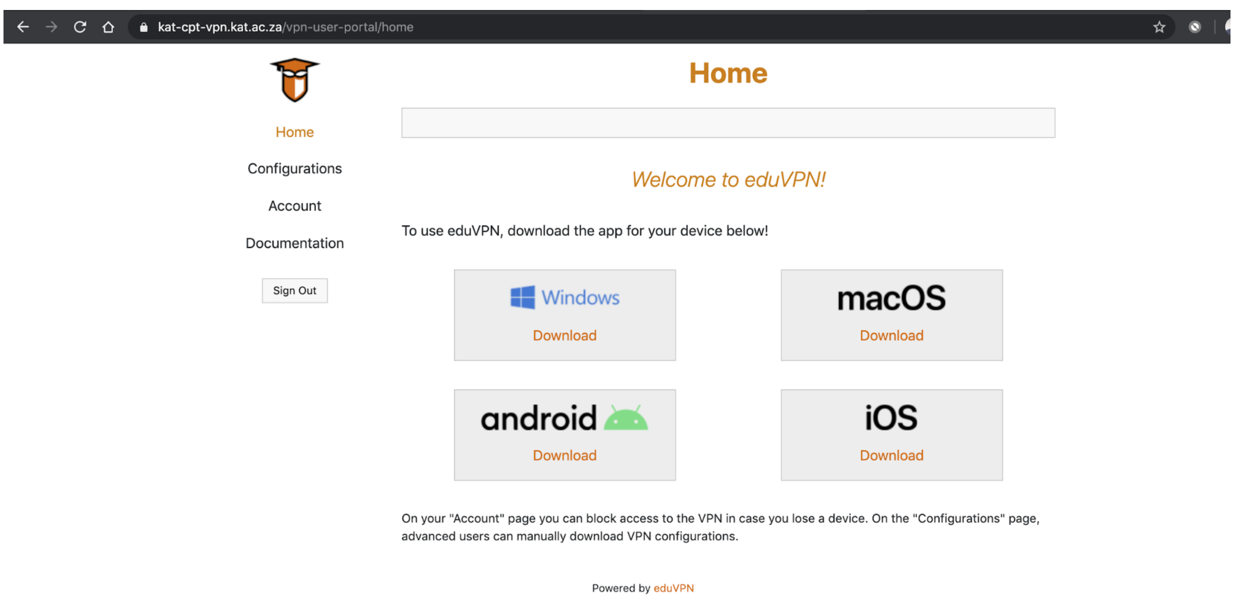


Figure B.8: MacOS EduVPN download page

Install and run the software (see **Figure** B.9).

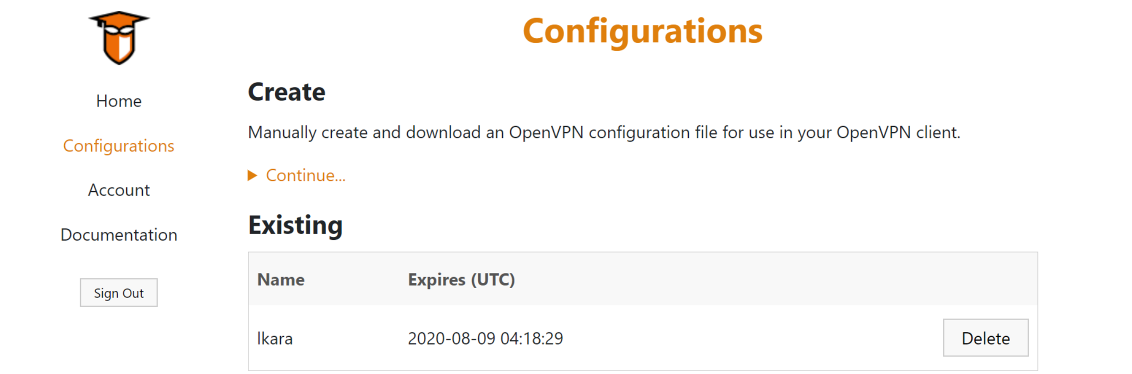


Figure B.9: MacOS EduVPN configuration file download

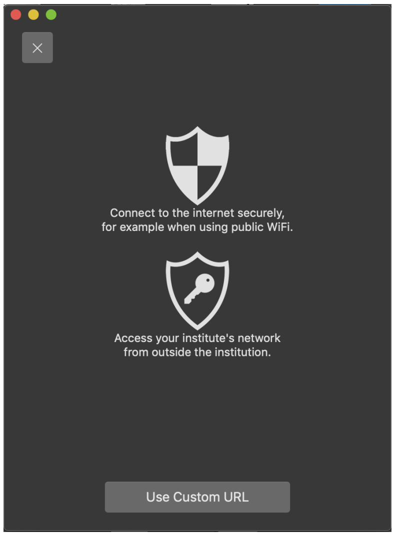


Figure B.10: MacOS EduVP custon url dialoque box

Click "Use Custom URL" (see **Figure** B.10)
Add `https://kat-cpt-vpn.kat.ac.za` to the "Enter URL" as in **Figure** B.11

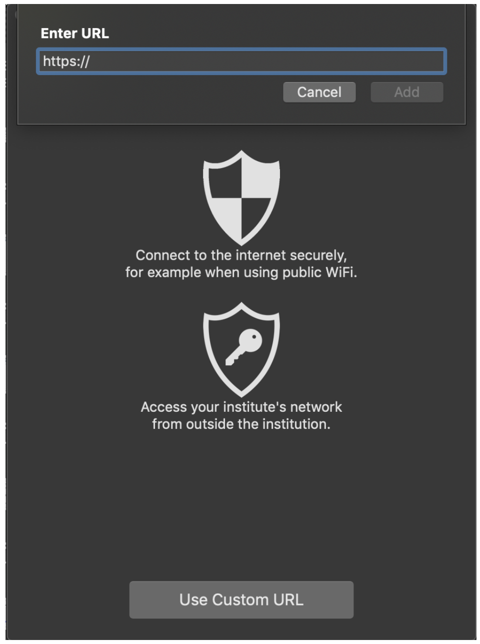


Figure B.11: MacOS EduVP custon url dialoque box

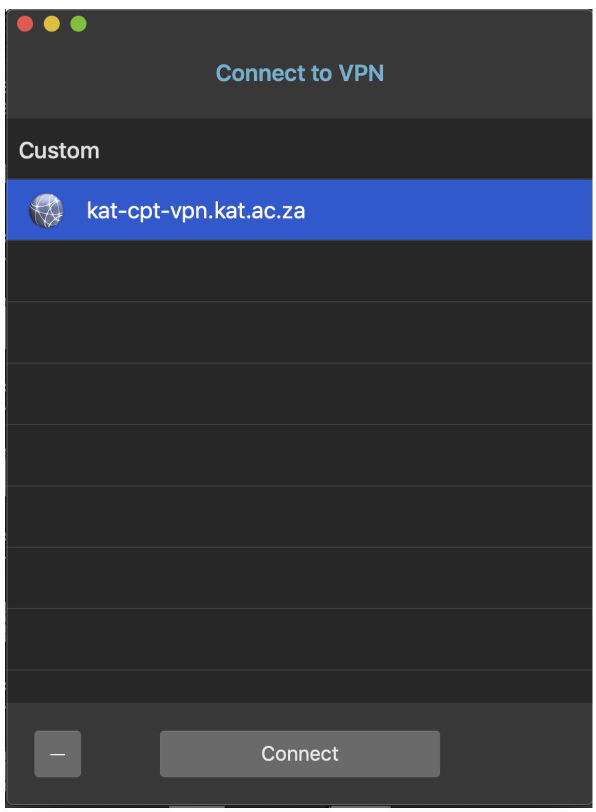


Figure B.12: Connect to EduVPN dialogue

Press connect (see **Figure** B.12).
Once connected to VPN go to https://ipa.kat.ac.za/ipa/ui to set a permanent and secure

password for yourself that you will remember. If you already have an account but eduVPN couldn't connect
Go to https://kat-cpt-vpn.kat.ac.za/vpn-user-portal/account and press Revoke (see **Figure** B.13).

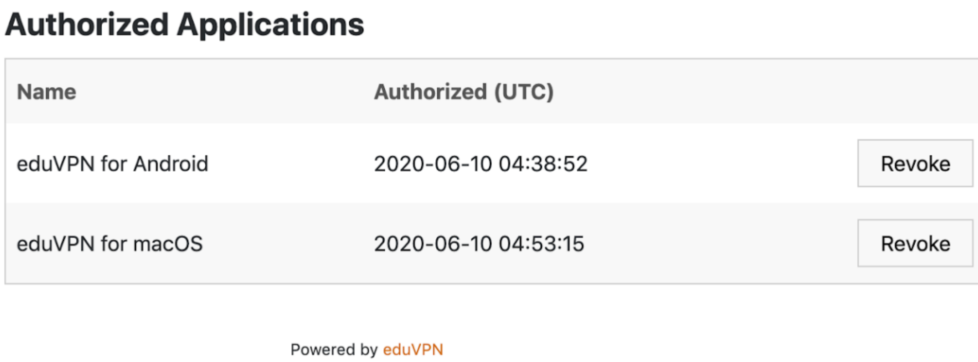


Figure B.13: Authorized Application

After pressing connect. The message as shown in **Figure** B.14 will be displayed.

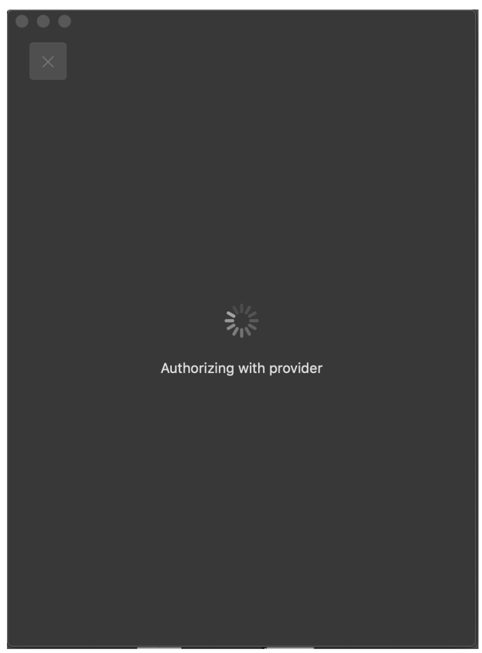


Figure B.14: MacOS EduVPN connection in progress

The window as shown in **Figure** B.15 will pop-up and press "Approve application"

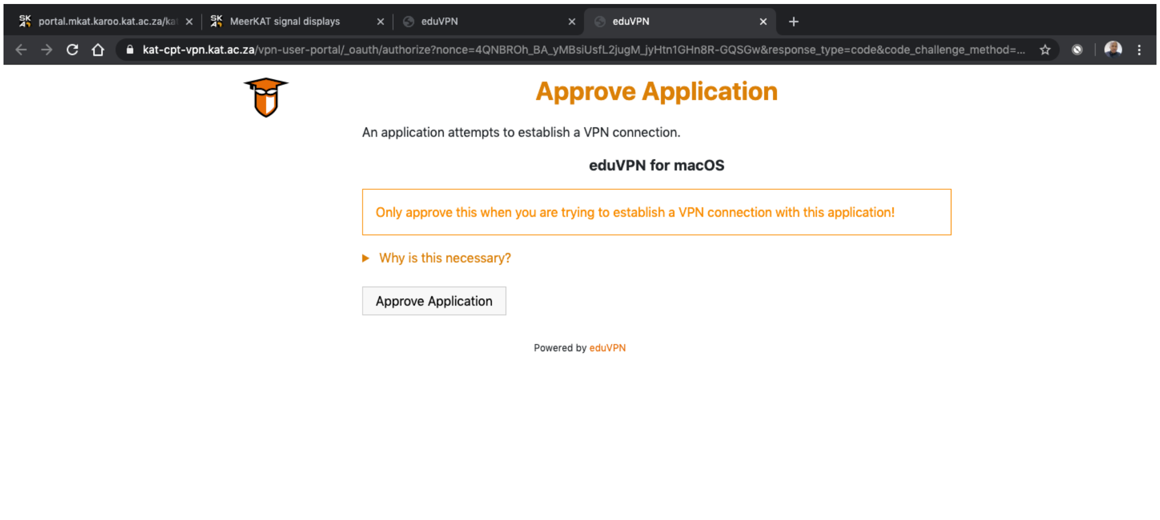


Figure B.15: MacOS EduVPN aprrove application dialogue

Click connect after approving and the you will be connected

B.3 Procedure for Windows

The procedure for connecting to VPN on Windows is more or less the same as that for macOS.

Download the EduVPN Software for Windows from https://kat-cpt-vpn.kat.ac.za/vpn-user-portal/home (see **Figure** B.16)

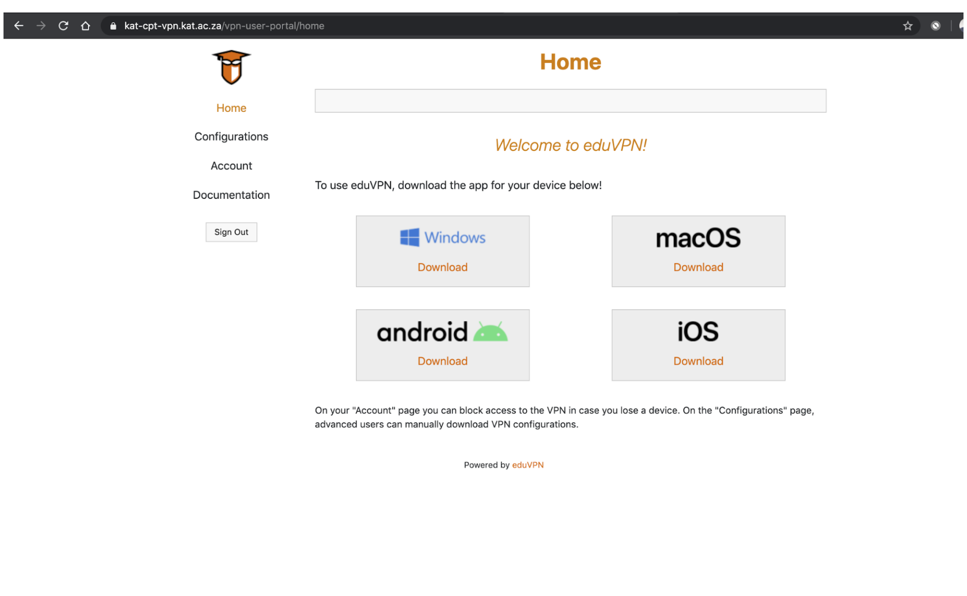


Figure B.16: MacOS EduVPN home page

The link to the application will be in your downloads folder as shown in **Figure** B.17, click to install. Once the installation is done, you can close the installer and click on the

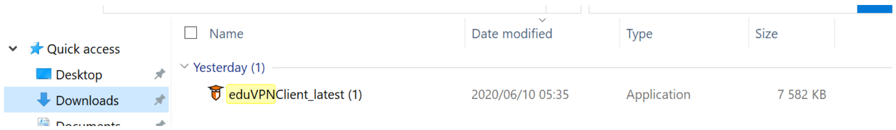


Figure B.17: Windows EduVPN installation

EduVPN item in your start menu (see **Figure** B.18) to launch VPN

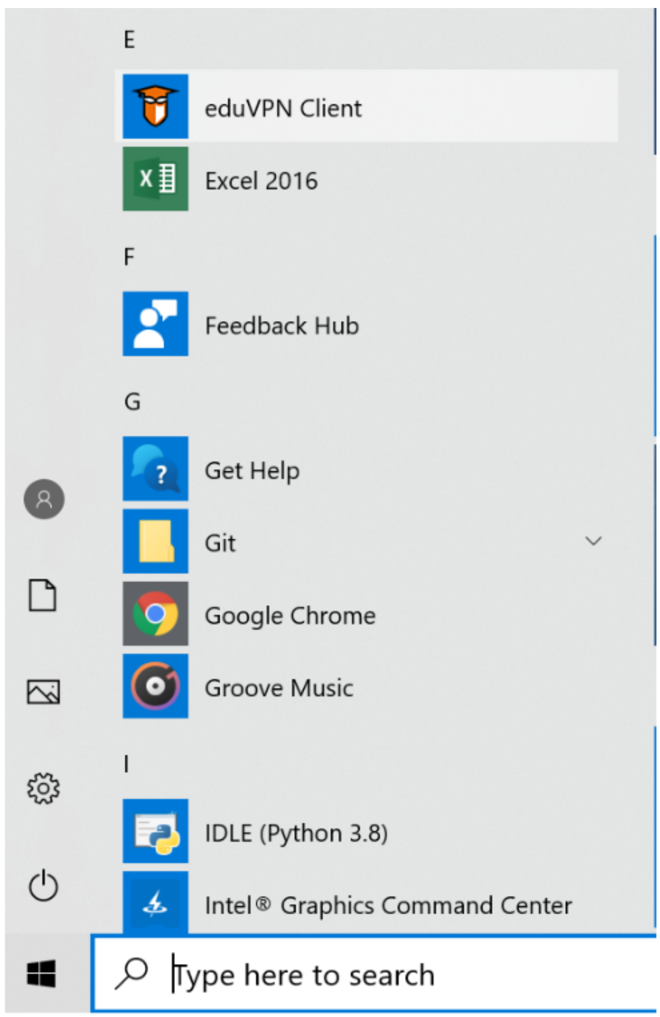


Figure B.18: Windows EduVPN client launch

You will be asked how you would like to use VPN, choose 'Add other address' as shown in **Figure** B.19 .

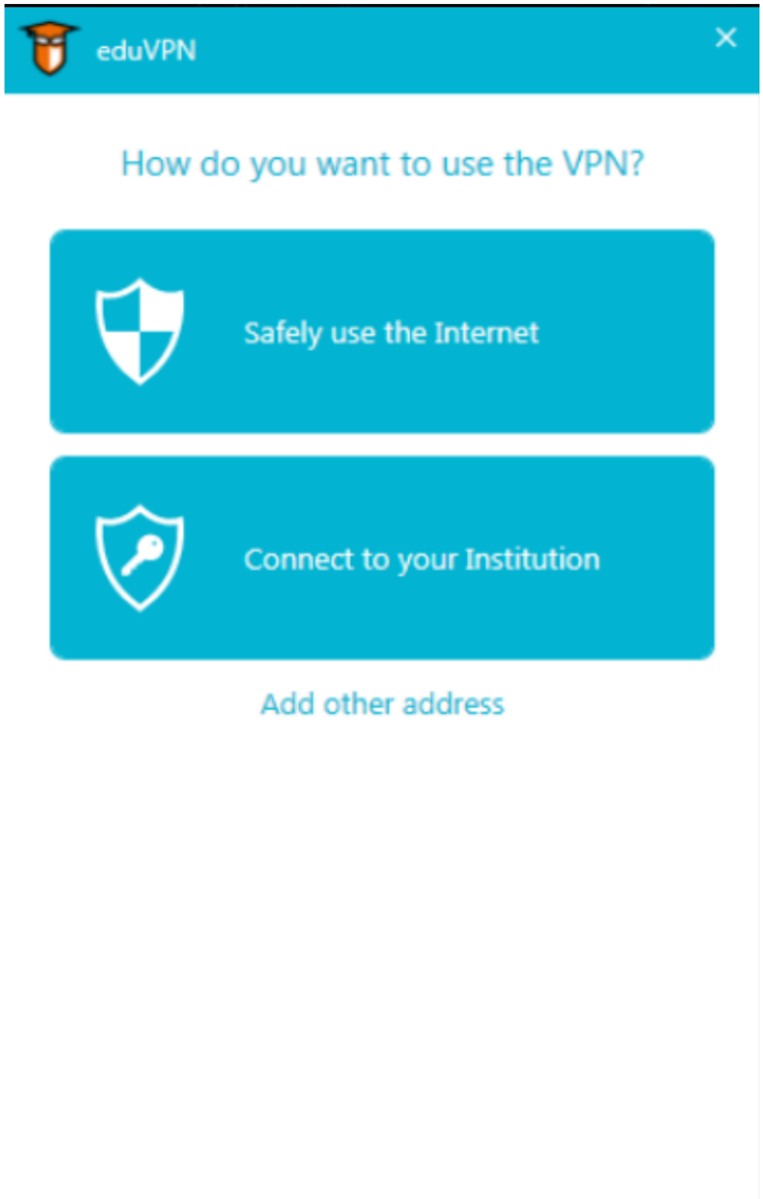


Figure B.19: Windows EduVPN add address option

kat-cpt-vpn.kat.ac.za/ is the URL of our provider, click 'connect!' (as in **Figure** B.20) once you are done.

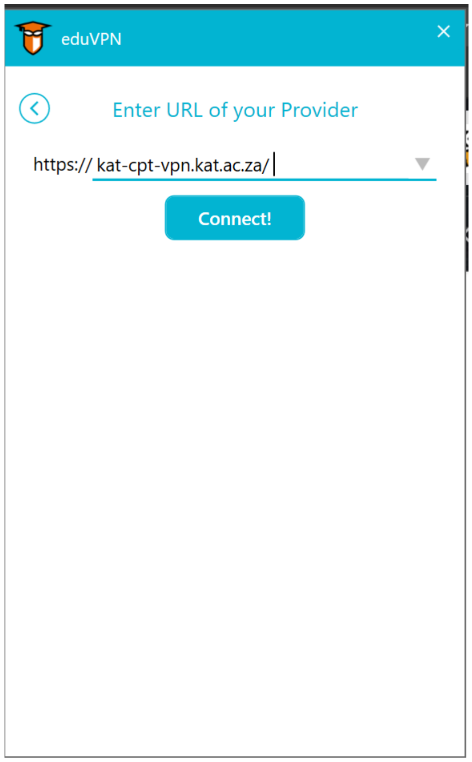


Figure B.20: Windows EduVPN installation enter url dialogue box

Click 'SARAO Staff access' as shown in **Figure** B.21. You will be directed to the Edu-

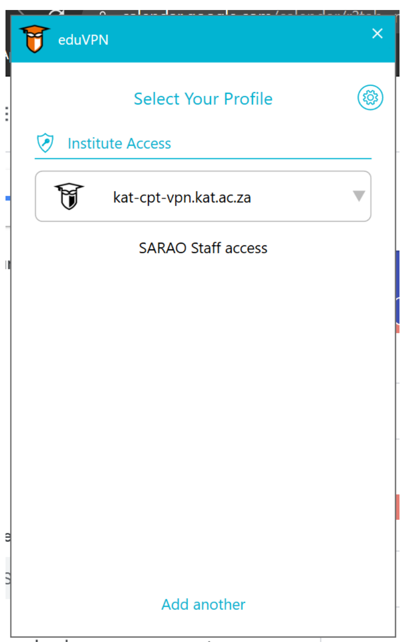


Figure B.21: Windows EduVPN SARAO staff access

VPN site where you will be required to approve the application like the way you would for macOS. After clicking ‘Approve Application’, you should be connected to VPN as shown in **Figure** B.22

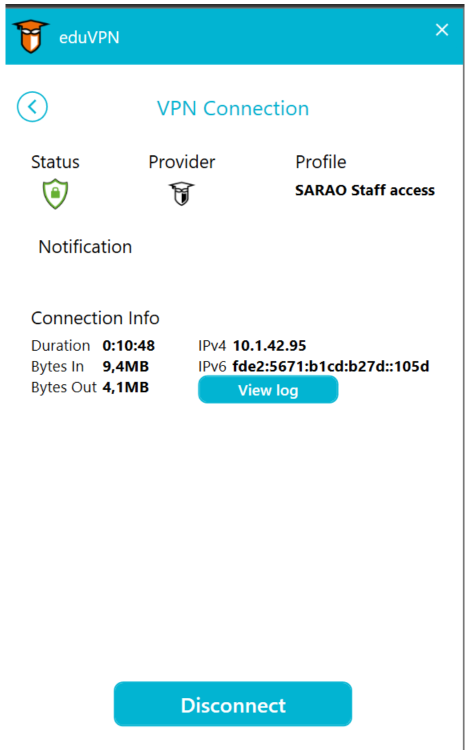


Figure B.22: Windows EduVPN connected

Everytime you want to connect to EduVPN, just click on the item in the start menu as shown in step 3, there will be no need to enter the URL of the provider. It should automatically appear, all you have to do is repeat step 6. Don’t forget to remove the old application every time you install a new one by clicking ‘Revoke’ (see **Figure** B.23) on the EduVPN site.

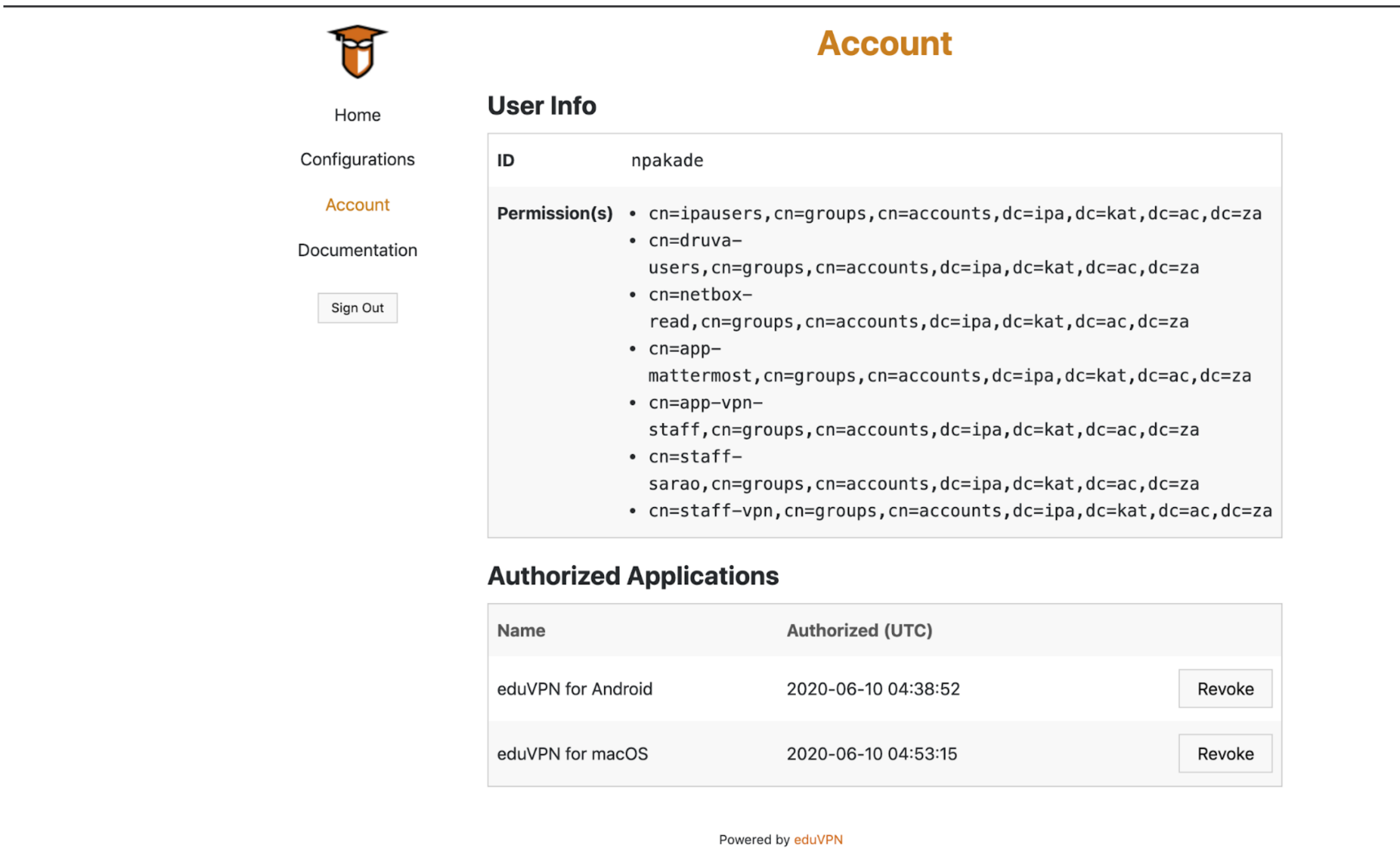


Figure B.23: Windows revoke EduVPN application

Bibliography

[1] Thomas Bennett. *Science Process Requirement Specification*. SARAO internal document, 2018.

[2] Hendrik Bester. *Antenna Positioner Requirement Specification*. SARAO internal document, 2013.

[3] Johan Burger. *TFR System Requirement Specification*. SARAO internal document, 2018.

[4] ComRad. Comrad manual, 2018.

[5] Sias Malan. *Digitiser System Requirement Specification*. SARAO internal document, 2017.

[6] NIST. Si unit rules and style conventions, 1984.

[7] Isak Theron. *Receiver System Requirement Specification*. SARAO internal document, 2015.

[8] Tyrone van Bella. *Correlator Beamformer Requirement Specification*. SARAO internal document, 2017.

[9] Lize van Heveer. *CAM subsystem Requirements specification*. SARAO internal document, 2018.