## **WELLIMOS NOTES**

### server: wellimos.niwa.co.nz

talk to IT about access, they'll require a copy of your public key.

**etc/cron.d/nzodn-import-moorings**

* This sits in cron.d so executes every minute? calls the mooring data ingestions scripts, command line argument is configuration file for data source (niwa ftp info) and destination (nzodn DB info), see below.

#Schedule a daily import for mooring data. Get don't expect new data

#most days, but when there is data we want it imported as soon as possible

2 10 \* \* \* robot /home/robot/aodn/import/mooring/bin/loadAllMooringData.sh -c /home/robot/aodn/import/mooring/config/prod.ini >> /tmp/nzodn-mooring-import.log

**/home/robot/aodn/import/mooring/config/prod.ini**

* Configuration file for data source (niwa ftp info) and destination (nzodn DB info)

[source]

type=ftp

server=ftp.niwa.co.nz

directory=incoming/AODN/Moorings\_data

fileglob=\*\.DAT3

user=anonymous

password=ITSystemsDevelopmentTeam@niwa.co.nz

[aodn]

directory=/data/niwa/publish/mooring

server=localhost

database=nzodn

user=nzodn\_admin

password=QuRUtnA9KfqFLOy2bToZ

**/tmp/nzodn-mooring-import.log**

* Not very helpful

[2020/01/13 10:02:01] INFO : Fetching data

[2020/01/14 10:02:01] INFO : Fetching data

**/home/robot/aodn/import/mooring/bin/mooringASCIIDef**

* Contains mooring data variable definitions

BEGIN {

/\* static lookups for data fields and flags, this is part of the format definition \*/

DATAFIELDS["tem"]="temperature";

DATAFLAGS["tem"]="hasTemperature";

DATAFIELDS["con"]="conductivity";

DATAFLAGS["con"]="hasConductivity";

DATAFIELDS["dir"]="currentDirection";

DATAFLAGS["dir"]="hasCurrentDirection";

DATAFIELDS["dirm"]="currentDirectionMagnetic";

DATAFLAGS["dirm"]="hasCurrentDirectionMagnetic";

DATAFIELDS["dirt"]="currentDirectionTrue";

DATAFLAGS["dirt"]="hasCurrentDirectionTrue";

DATAFIELDS["spe"]="currentSpeed";

DATAFLAGS["spe"]="hasCurrentSpeed";

DATAFIELDS["oxy"]="dissolvedOxygen";

DATAFLAGS["oxy"]="hasDissolvedOxygen";

DATAFIELDS["oxp"]="oxygenSaturation";

DATAFLAGS["oxp"]="hasOxygenSaturation";

DATAFIELDS["pre"]="pressure";

DATAFLAGS["pre"]="hasPressure";

DATAFIELDS["prea"]="pressureAbsolute";

DATAFLAGS["prea"]="hasPressureAbsolute";

DATAFIELDS["prer"]="pressureRelative";

DATAFLAGS["prer"]="hasPressureRelative";

DATAFIELDS["sal"]="salinity";

DATAFLAGS["sal"]="hasSalinity";

DATAFIELDS["cdep"]="depthInterpolated";

DATAFLAGS["cdep"]="hasDepthInterpolated";

}

**/home/robot/aodn/import/mooring/bin/loadAllMooringData.sh**

* Master ingestion script
* Download all .DAT3 files

from: [ftp.niwa.co.nz/incoming/AODN/Moorings\_data/](ftp://ftp.niwa.co.nz/incoming/AODN/Moorings_data/)

destination: /data/niwa/publish/mooring

* Prepare the directory structure for each data file as published on the AODN file server
* Check if file exists in data folder, if it does, ignore it, report message
* Calls loadMooringData using nzodn server configuration details from prod.ini

$BIN/loadMooringData.sh -s $aodnServer -d $aodnDatabase -u $aodnUser -p $aodnPassword -f $fileurl $([ $verbose -eq 1 ] && echo "-v")

* Report errors

**/home/robot/aodn/import/mooring/bin/loadMooringData.sh**

* Load marine measurement data for fixed mooring stations stored in NIWA's ASCII format connects to postgres server executes a bunch on SQL commands

print "INSERT INTO instrument (name, description, serial\_number)";

print "SELECT \* FROM (";

print " VALUES (|"name"|,|"desc"|,|"serial"|)";

print ") AS m";

print "WHERE NOT EXISTS (";

print " SELECT TRUE FROM instrument WHERE name = |"name"| AND serial\_number = |"serial"|";

* load location record, if needed
* loads the datasource and its data set

## PostgresSQL

### Connect to server localhost

$psql -h localhost -d nzodn -U nzodn\_admin

**Password for user nzodn\_admin:** QuRUtnA9KfqFLOy2bToZ

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

------------+-------------+----------+-------------+-------------+-----------------------

aodn\_ctd | aodn\_admin | UTF8 | C.UTF-8 | C.UTF-8 |

geonetwork | postgres | UTF8 | en\_NZ.UTF-8 | en\_NZ.UTF-8 |

nzodn | nzodn\_admin | UTF8 | C.UTF-8 | C.UTF-8 |

postgres | postgres | UTF8 | C.UTF-8 | C.UTF-8 |

ss\_nzodn | postgres | UTF8 | en\_NZ.UTF-8 | en\_NZ.UTF-8 |

template0 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | C.UTF-8 | C.UTF-8 | =c/postgres