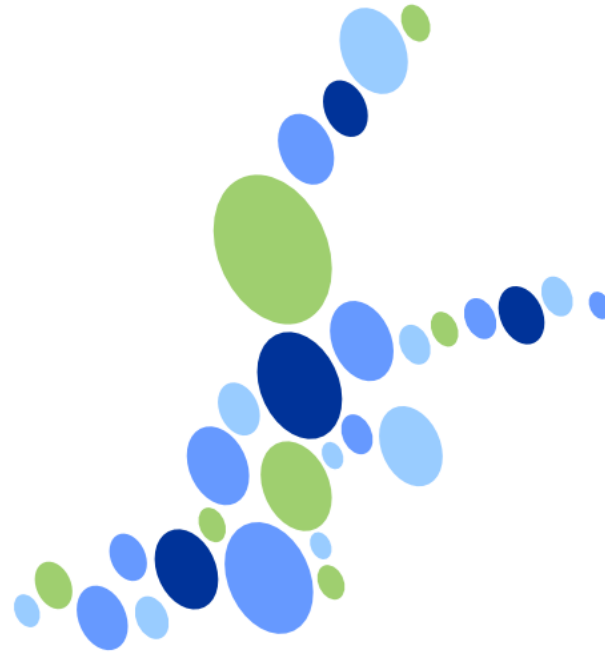


# baltrad



baltrad



# DIY: baltrad installation for beginners

*On Ubuntu 12.04*

*Jesper Ellerbæk Nielsen  
Aalborg University, DK*

# Required steps for node installation

- Prepare your machine for BALTRAD – download and install required software
- Create a Postgre SQL database and database user
- Download the BALTRAD node software
- Install BALTRAD node software
- Add ODIM source definitions
- Get your own data into the node



# Prepare your machine – download and install required software

You can use the 'apt-get' package manager to download and install the needed software.  
(apt-get is a simple command line interface for downloading and install packages)

- Postgre SQL
- Java SDK
- Gcc/g++
- Gnu make
- autoconf
- bzip2
- Doxygen
- Libpng-dev
- Lipfreetype6
- Git-core
- Patch
- Open ssl
- libicu



# Prepare your machine – download and install required software

Make sure that your package-list is up to date:

```
$ sudo apt-get update
```

Install Postgres SQL (Client and server):

```
$ sudo apt-get install postgresql-9.1 postgresql-client-9.1 postgresql-server-dev-9.1
```

Install Java: `$ sudo apt-get install openjdk-6-jre openjdk-6-jdk`

Install autoconf: `$ sudo apt-get install autoconf`

Install make: `$ sudo apt-get install make`

Install gcc: `$ sudo apt-get install gcc`

Install g++: `$ sudo apt-get install g++`

Install patch: `$ sudo apt-get install patch`

Install libc6: `$ sudo apt-get install libc6 libc6-dev`

Install bzip2: `$ sudo apt-get install bzip2`

Install doxygen: `$ sudo apt-get install doxygen`

Install libpng : `$ sudo apt-get install libpng-dev`

Install libfreetype: `$ sudo apt-get install libfreetype6 libfreetype6-dev`

Install git-core: `$ sudo apt-get install git-core`

Install libicu: `$ sudo apt-get install libicu-dev`

Install Open ssl: `$ sudo apt-get install openssl libssl-dev dpkg-dev`



# Create a Postgre SQL database and database user

- Change the security level a bit
- Create a database and database user for the node



# Create a Postgre SQL database and database user

Change the security level a bit

Log in as root:

```
$ su
```

If needed you can add password to the root account to allow terminal root logins by:

```
$ sudo passwd root
```

Use e.g. the vi editor to change the content of pg\_hba.conf file

```
# vi /etc/postgresql/9.1/main/pg_hba.conf
```

The end of the file should look like this:

```
# TYPE  DATABASE  USER  CIDR-ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local  all      all      md5
# IPv4 local connections:
host   all      all      127.0.0.1/32  md5
# IPv6 local connections:
host   all      all      ::1/128      md5
```

Use: arrows to move around, 'r' to replace a character, 'x' to delete a character,

'w' to write the file, 'q' to quit the editor.

See for more commands if needed : [www.cs.rit.edu/~cslab/vi.html](http://www.cs.rit.edu/~cslab/vi.html)



# Create a Postgre SQL database and database user

Create a database and database user for the node

After editing the of pg\_hba.conf the server need to be restarted:

```
# /etc/init.d/postgresql restart
```

Create database and user:

```
Log on as postgres: # su - postgres
```

```
$ psql
```

```
postgres=# create user baltrad with password 'baltrad';
```

```
postgres=# create database baltrad with owner baltrad;
```

You can check what you have made by:

```
postgres=# \l
```

If everything looks fine quit by:

```
postgres=# \q
```

```
$ exit
```

```
# exit
```





# Download the baltrad node software

Change the ownership of the /opt folder from root to your user-profile

```
$ sudo chown -hR baltrad /opt
```

 in this case baltrad is the user

Create a folder for the node installer software

```
$ mkdir /opt/software
```

```
$ cd /opt/software
```

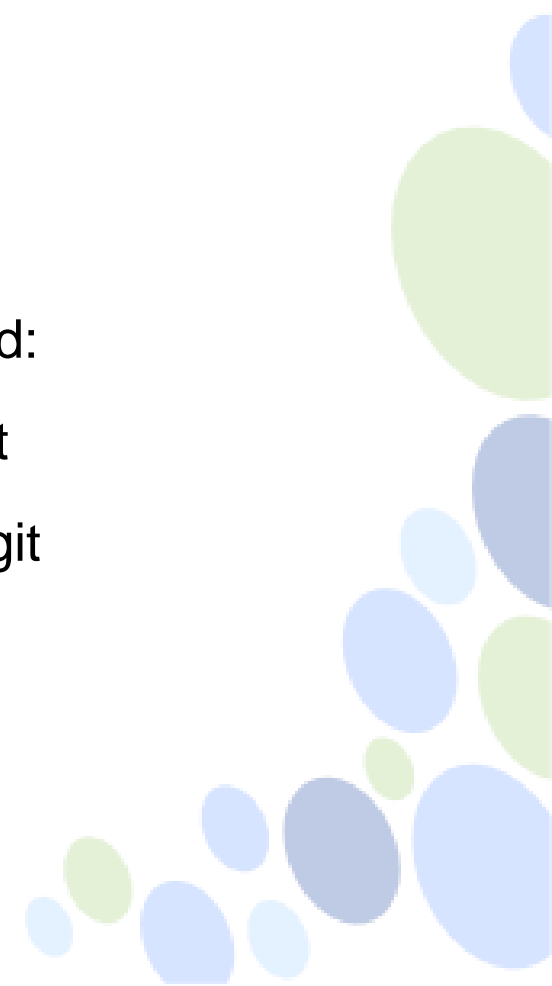
Downloading the BALTRAD software are now straight forward:

```
/opt/software$ git clone git://git.baltrad.eu/node-installer.git
```

```
/opt/software$ git clone http://git.baltrad.eu/node-installer.git
```

Go to the node-installer folder:

```
$ cd node-installer
```



# Install the BALTRAD node software

Installing baltrad is a one line command

See your options by typing `./setup --help`

`./setup`

`--nodename=your.node.name`

`--prefix=/opt/baltrad/`

`--jdkhome=/usr/lib/jvm/java-6-openjdk-amd64/`

`--with-psql=/usr/include/postgresql,/var/lib/postgresql/9.1/`

`--bdb-auth=noauth`

`--bdb-port=8090`

(default)

`--tomcatport=8080`

(default) This port needs to be open towards the outside world

`--tomcatpwd=pwd2tomcat`

`--keystorepn=yes`

(default)

`--keystorepwd=pwd2tomcat`

(must be the same as `--tomcatpwd`)

`--tomcatsecureport=8443`

(default) This port needs to be open towards the outside world

`--gitrepo=http://git.baltrad.eu`

(default)

`--experimental`

(needed on Ubuntu)

`--with-rave`

(optional)

`--with-rave-gmap`

(optional)

`--with-beamb`

(optional)

`--with-bufr`

(optional)

`--with-bropo`

(optional)

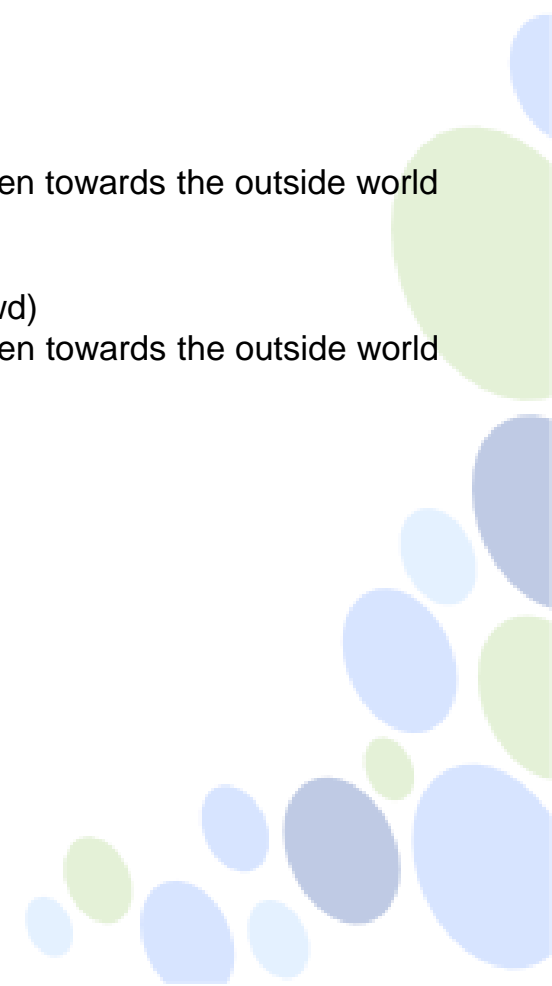
`--with-bdbfs`

(optional)

Install

`--excludedb`

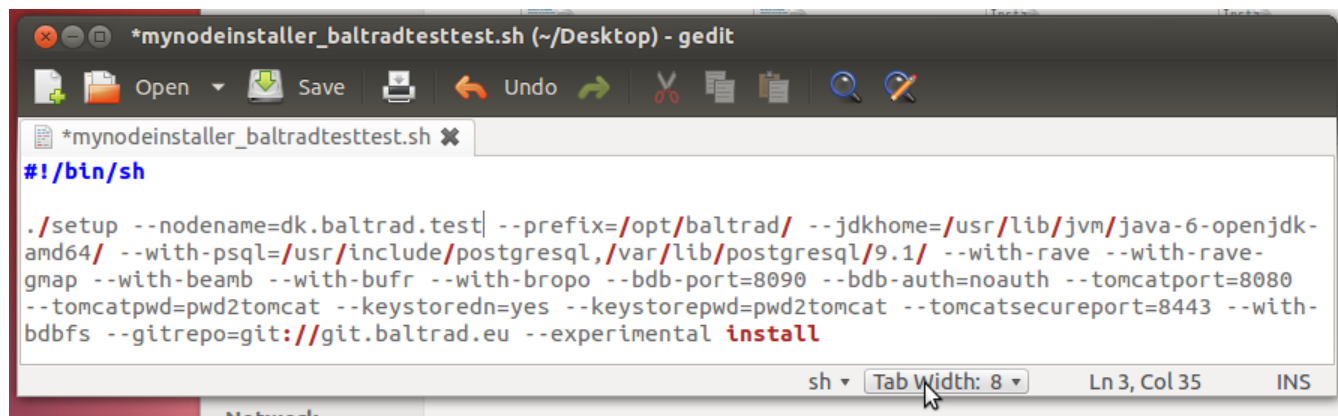
`--reinstalldb`



# Install the BALTRAD node software

You might want to make a installation script for later use

Use for example Gedit to create you script:



The screenshot shows a Gedit window titled '\*mynodeinstaller\_baltradtesttest.sh (~/Desktop) - gedit'. The script content is as follows:

```
#!/bin/sh

./setup --nodename=dk.baltrad.test| --prefix=/opt/baltrad/ --jdkhome=/usr/lib/jvm/java-6-openjdk-
amd64/ --with-psql=/usr/include/postgresql,/var/lib/postgresql/9.1/ --with-rave --with-rave-
gmap --with-beamb --with-bufr --with-bropo --bdb-port=8090 --bdb-auth=noauth --tomcatport=8080
--tomcatpwd=pwd2tomcat --keystoredn=yes --keystorepwd=pwd2tomcat --tomcatsecureport=8443 --with-
bdbfs --gitrepo=git://git.baltrad.eu --experimental install
```

The status bar at the bottom of the window indicates 'sh', 'Tab Width: 8', 'Ln 3, Col 35', and 'INS'.

If your script is executable and located together with the node-installer, you can run your 'customized' installation script by:

```
$ /opt/baltrad/software/node-installer $./myinstallscript.sh
```

# Install the BALTRAD node software

During the installation you will be asked to enter name and other details:

```
baltrad@baltrad-ThinkPad-T400: /opt/software/node-installer
apache-ant-1.8.0/lib/ant-trax-1.8.0.pom.sha512
apache-ant-1.8.0/lib/ant-trax.jar
apache-ant-1.8.0/lib/ant.jar
apache-ant-1.8.0/lib/libraries.properties
apache-ant-1.8.0/lib/xercesImpl.jar
apache-ant-1.8.0/lib/xml-apis.jar
creating private key in /opt/baltrad/etc/bltnode-keys/dk.baltrad.testtest.priv
exporting public key to /opt/baltrad/etc/bltnode-keys/dk.baltrad.testtest.pub
creating keystore file /opt/baltrad/etc/bltnode-keys/keystore.jks
What is your first and last name?
[Unknown]: Jesper Nielsen
What is the name of your organizational unit?
[Unknown]: Dept of Civil Engineering
What is the name of your organization?
[Unknown]: Aalborg University
What is the name of your City or Locality?
[Unknown]: Aalborg
What is the name of your State or Province?
[Unknown]: Aalborg
What is the two-letter country code for this unit?
[Unknown]: dk
Is CN=Jesper Nielsen, OU=Dept of Civil Engineering, O=Aalborg University, L=Aalbo
rg, ST=Aalborg, C=dk correct?
[no]:
```

When you get this message the installation is done

```
baltrad@baltrad-ThinkPad-T400: /opt/software/node-installer
Using CATALINA_TMPDIR: /opt/baltrad/third_party/tomcat/temp
Using JRE_HOME: /usr/lib/jvm/java-6-openjdk-amd64/
Using CLASSPATH: /opt/baltrad/third_party/tomcat/bin/bootstrap.jar

===== SUCCESS =====
Your BDB sources might not be up-to-date. You can import them from
Rave's radar-db with the following command:

/opt/baltrad/baltrad-db/bin/baltrad-bdb-client \
  import_sources \
  --url=http://localhost:8090 \
  --dry-run \
  /opt/baltrad/rave/config/odim_source.xml

You can omit some changes by adding '--ignore=src' to the command.
Once you are satisfied with what the importer will do, omit the
'--dry-run' switch and let it work on the actual database.

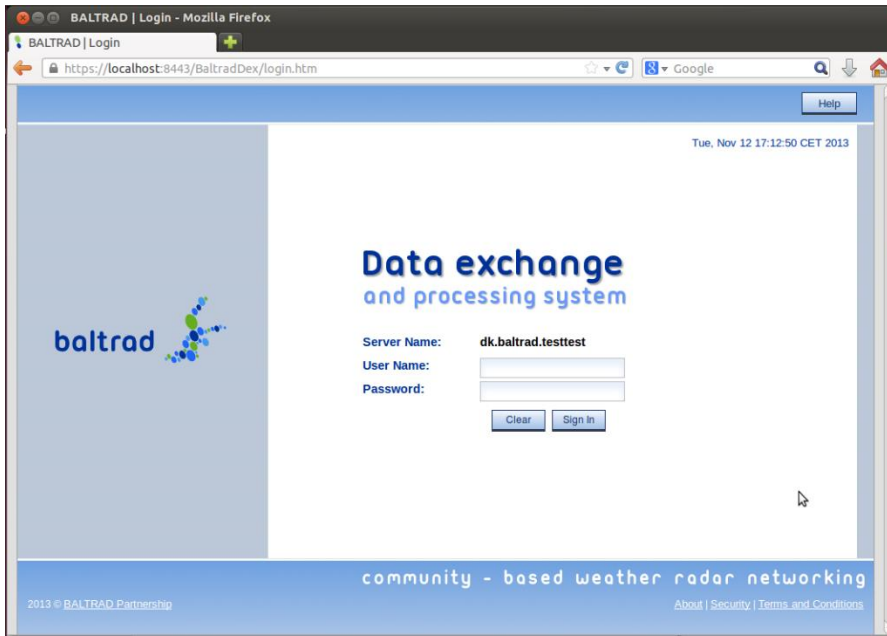
If you are planning to use any specific binary from a subsystem you
might have to setup your environment so that it is properly
configured. An easy way to setup the environment is to source
/opt/baltrad/etc/bltnode.rc

baltrad@baltrad-ThinkPad-T400: /opt/software/node-installer$
```

# Check that your node is running

Use the browser to view the baltrad web interface (<http://localhost:8080>)

– if you get this it works:



You can also check node status, start, stop your node by eg.

```
$ /opt/baltrad/bin/./bltnode --all status
```

```
$ /opt/baltrad/bin/./bltnode --all start
```

```
$ /opt/baltrad/bin/./bltnode --all stop
```

The default username and password is 'admin' and 'baltrad'

# Add ODIM source definitions

The weather radar identifiers need to be added manually – however the xml file containing the information comes with the node software.

First you need to setup the environment by:

```
$ source /opt/baltrad/etc/bltnode.rc
```

Furthermore the odim\_source.xml needs to be executable - if not:

```
/opt/baltrad/rave/config$ chmod +x odim_source.xml
```

To add the sources, use the baltrad-bdb-client:

```
$ /opt/baltrad/baltrad-db/bin/baltrad-bdb-client \  
import_sources \  
--url=http://localhost:8090 \  
/opt/baltrad/rave/config/odim_source.xml
```



# Get your own data into the node

The simplest way of injecting your own data into your node is by the 'n2b' injector.

- The radar data must be in ODIM\_H5 format.

'n2b' need two directories: a working directory and a data directory.

Create default working directory:           \$ mkdir /opt/baltrad/n2b

Create default data directory:           \$ mkdir /opt/baltrad/n2b/data

Activating the injector with default paths:   \$ n2b

Deactivating the injector:           \$ n2b --kill

Information and help:           \$ n2b --help

You might need to setup the environment by:   \$ source /opt/baltrad/etc/bltnode.rc

Location of 'n2b': /opt/baltrad/rave/bin

