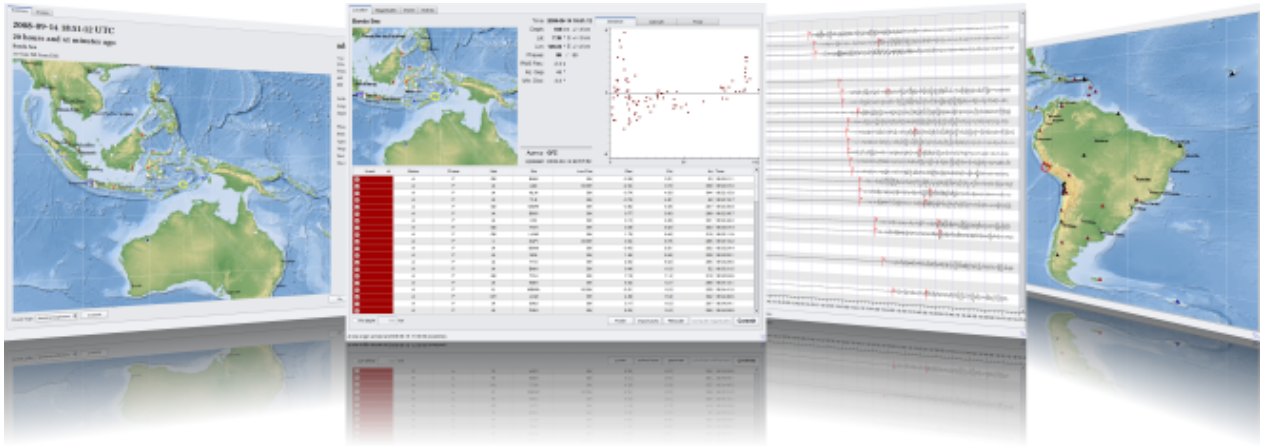
gempa

SeisComP installation

Installation Manual

Prepared for:	GEMPA (Potsdam), Germany	
Prepared by:	gempa GmbH	
Issued by:	E. Ellguth, R. Henneberger	
Reference:	gempa _installation_manual_gempa	Issue: 1.0
Date:	June 11, 2024	

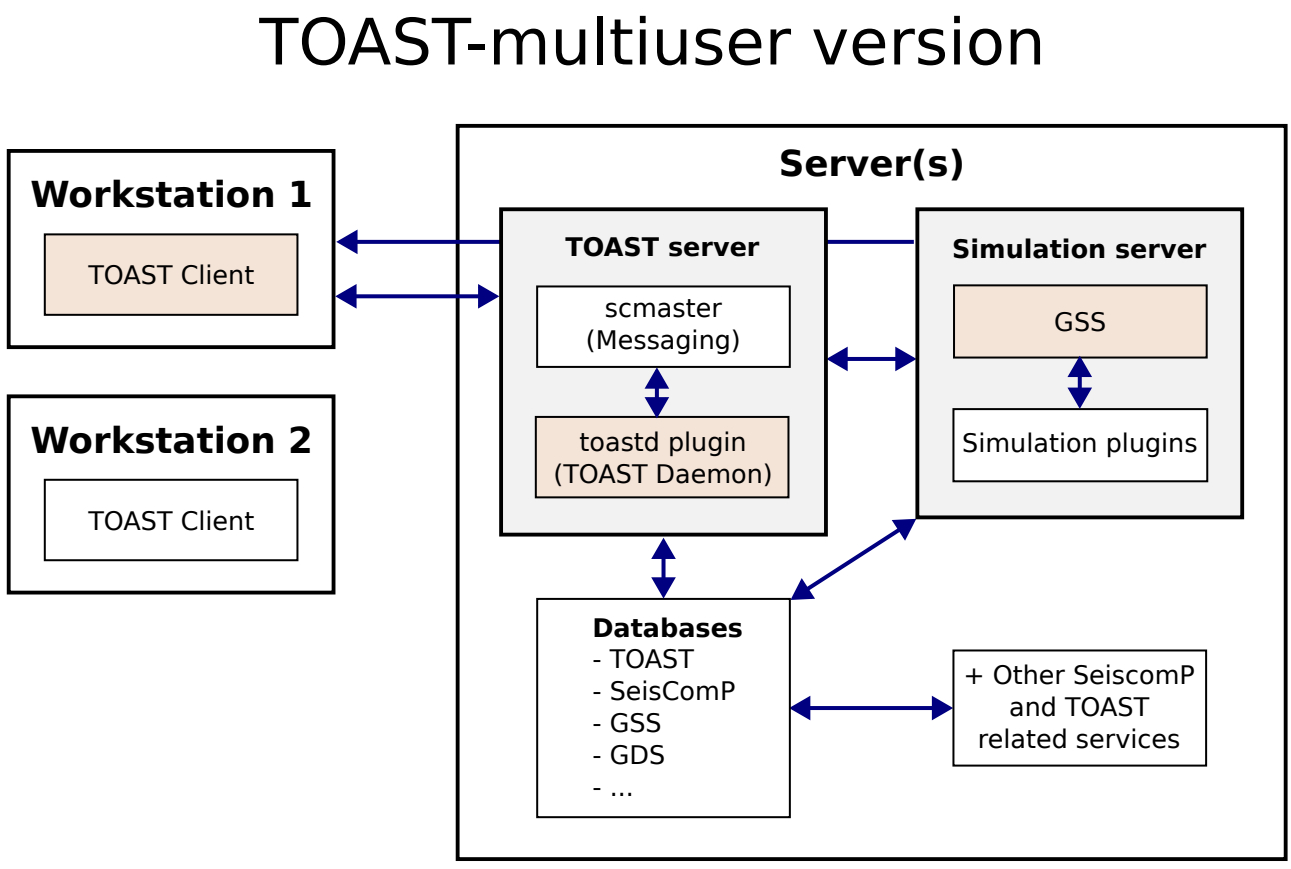


Contents

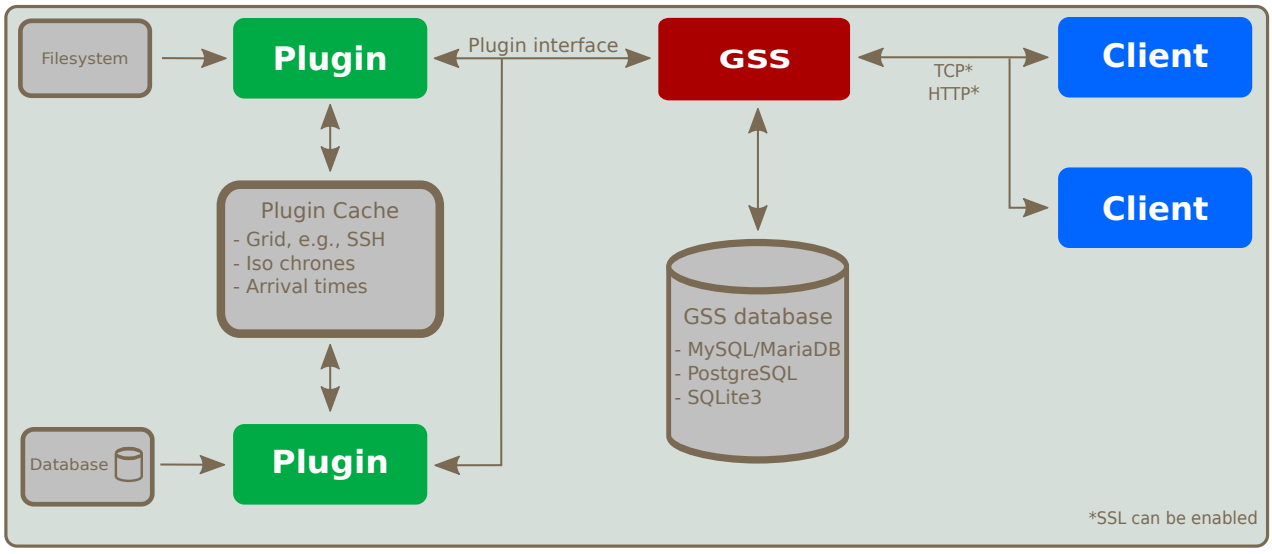
Chapter 1

Architecture Overview

1.1 TOAST



1.2 GSS



Chapter 2

OS Installation and Configuration

2.1 System Update

Before going ahead with the installation it is very important to install available fixes and security updates. RHEL uses the package manager apt to maintain packages. Run the command below to run a system-wide update. Keep in mind that it is a good idea to keep the system always up-to-date to avoid damage.

1. Login as root

```
root@host:~# apt install screen vim mc wget net-tools telnet
```

2. Start screen session

```
root@host:~# screen
```

3. Install the updates

```
root@host:~# apt update
root@host:~# apt upgrade
```

2.2 SSH

2.2.1 Configuration

The SSH service reads the system-wide configuration data from file /etc/ssh/sshd_config. The configuration supports several options which can be given line by line as key value pairs. The default configuration tries to resolve the remote client's IP which may cause login delays. To disable this feature set the option UseDNS to 'no'.

1. Login as root
2. Open the file */etc/ssh/sshd_config* in a text editor and add/replace the default settings by the following lines:

```
UseDNS no
```

2.2.2 X11 Forwarding

By default X11 Forwarding is not supported by the minimal installation of OS. To be able to open graphical user interfaces as for example scconfig remotely, X11 forwarding should be enabled through the following steps.

2.2.2.1 Configuration

1. Login as root
2. Install the required packages with:

```
root@host:~# apt install x11-xserver-utils xauth
```

3. Enable X11Forwarding in file /etc/ssh/sshd_config

```
X11Forwarding yes
#X11UseLocalhost no
```

4. The broken X11 forwarding error may also happen if the remote host where the SSH server is running has IPv6 support enabled. To fix this error open the /etc/ssh/sshd_config and uncomment line "AddressFamily all" (if any). Then add the line below to force the SSH server to use IPv4 only, but not IPv6.

```
AddressFamily inet
```

5. Reload the service configuration

```
root@host:~# systemctl reload sshd
```

6. Optional for testing:
7. Login as root

```
root@host:~# apt install xeyes
```

8. Logout
9. Connect as sysopto system:

```
sysop@host:~$ ssh -X sysop@newSystem
```

```
sysop@host:~$ xeyes
```

2.3 Core dumps

Core dumps can be used to analyze application crashes caused by segmentation faults. The dump may help the developer of the application to find out what went wrong. This section describes all neccessary configuration steps to enable this feature.

2.3.1 Configuration

1. Login as root

2. Open the file */etc/security/limits.conf* in a text editor and add the following line:

```
  sysop  soft core unlimited
```

3. Create a new directory where core dumps should be stored

```
root@host:~# install -m 777 -d /home/data/dumps
```

4. Change the file pattern of core dump in the file */etc/sysctl.conf* to

```
kernel.core_pattern = /home/data/dumps/core.%u.%e.%w.%s
```

```
kernel.core_uses_pid = 0
```

5. Apply the new configuration

```
root@host:~# ln -s ../sysctl.conf 99-sysctl.conf
```

6. Logout and login again to activate the changes

2.3.2 Configuration

Color highlighting reduces the risk of doing actions as root accidently which often results in ownership problems. The following steps reconfigure the system to use a green prompt for regular users and a red one for the superuser.

1. Login as root
2. Change into systemd configuration directory

```
root@host:~# cd /etc/profile.d/
```

3. Open the file *colors.sh* in a text editor and replace the default settings by the following lines:

```
#!/bin/bash
if [ $(id -u) -eq 0 >/dev/null 2>&1 ]; then
  export PS1='\[\033[01;31m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]# '
else
  export PS1='\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
fi
```

4. Save the changes

2.4 Screen

2.4.1 Configuration

1. Open the file */home/sysop/.screenrc* in a text editor and replace the default settings by the following lines:

```
# kill startupmessage:
startup_message off

# use no visual bell:
vbell off

# replace ctrl-A by ctrl-O
# escape ^Tt

# define a bigger scrollback, because default is 100 lines
```

```
defscrollback 8192

multiuser on

# statusline at the bottom of the terminal
#termcapinfo xterm "ks=E[?1lE:kuE[A:kd=E[B:kl=E[D:kr=E[C:kh=E[5~:kH=E[F"
# hardstatus alwayslastline "%{-b by}%-w%{+b yb}%50>%n %t%{-b by}%+w%< %=%d.%m. %c"
# Set the caption on the bottom line
#caption always "%{= kw}%-w%{= BW}%n %t%{-}%+w %-= @%H -%LD %d %LM -%c"
caption always "%{= kw}%-w%{= BW}%n %t%{-}%+w %-= @%H -%LD %d %M -%c"

# dynamic title: (entsprechend escape sequence [siehe PS1] oder bash[default])
#shelltitle '$ |bash'

# if ssh is dieing then screen should detach the sessions for itself:
autodetach on

# test some termcap stuff:
termcapinfo xterm*|rxvt* 'ti@:te@'


# let's screen try to use 'normal' $TERM:
term $TERM
```

2. Save the changes

2.5 NTP

With time the difference between system time and real time increases. The time shift is depending on the quality of the clock. The network time protocol (NTP) was designed to solve this issue and allows to synchronize system time with a remote time server.

2.5.1 Timezone configuration

When sharing data with other people it is strongly recommended to use the standard time(UTC) as system time to avoid misunderstandings. This subsection describes how to switch the system timezone to UTC.

1. Set timezone to UTC

```
root@host:~# dpkg-reconfigure tzdata
```

2.5.2 Installation

1. To install the NTP package, enter this

```
root@host:~# apt install ntp
```

2.5.3 Configuration

1. Login as root
2. Change to the system configuration directory

```
root@host:~# cd /etc
```

3. Open the file ntp.conf in a text editor and add one or more time server. For instance:

```
server 192.168.100.21
server 192.168.100.22
```

4. Save the changes and quit
5. Enable and restart the NTP serviceA

```
root@host:~# systemctl enable ntp
root@host:~# systemctl restart ntp
```

2.6 MariaDB

RHEL uses MariaDB as database server which is fully compatible with MySQL. The default installation uses a default password which should be changed to avoid security issues. The command line program mysql_secure_installation helps the user to secure the installation.

2.6.1 Installation

1. Login as root
2. Install MariaDB server

```
root@host:~# apt install mariadb-server mariadb-client
```

3. Start MariaDB

```
root@host:~# systemctl enable mariadb
root@host:~# systemctl start mariadb
```

4. Check the status of MariaDB

```
root@host:~# systemctl status mariadb
```

2.6.2 Configuration

1. Run the MariaDB secure installation script

```
root@host:~# mysql_secure_installation
```

2. Set the password for root account to mysqlroot

```
Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

3. Disable anonymous user logins

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
Remove anonymous users? [Y/n] Y
... Success!
```

4. Disable remote login for the root account

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
Disallow root login remotely? [Y/n] Y
... Success!
```

5. Remove the test database

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
Remove test database and access to it? [Y/n] Y
-Dropping test database...
... Success!
-Removing privileges on test database...
... Success!
Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
```

6. Reload privilege tables

```
Reload privilege tables now? [Y/n] Y
... Success!
Cleaning up...
All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.
Thanks for using MariaDB!
```

By default the MariaDB service only listens to connections originating from the same machine. If you plan to distribute the SeisComP modules and/or GUIs on different machines you need to allow remote network connections.

Change bind-address in file /etc/my.cnf.d/server.cnf

```
[mysqld]
bind-address = 0.0.0.0
```

The performance of the database server can be improved significantly by

- Increase the memory pool size. Test the default before making the change:

```
mysql -u root -p
   show variables like 'innodb_buffer_pool_size';
```

  The optimum **buffer_pool_size** depends on your system (RAM size) and only needs to be set if required.
  - Recommended value: `buffer_pool_size = 512M`
  - Minimum value: `buffer_pool_size = 64M`
- Reducing the database hard drive synchronization. Note: This configuration may result in the loss of up to 1 s of data in case of hard machine crash. Nevertheless, if SeisComP is the only application using the database, this setting is considered a justifiable trade off between reliability and performance.

Add database optimizations in /etc/my.cnf.d/server.cnf

```
[mysqld]
innodb_buffer_pool_size = <new value>
innodb_flush_log_at_trx_commit = 2
```

Restart MariaDB database server to apply changes:

```
root@host:~# systemctl restart mariadb
```

Add MariaDB service to autostart with:

```
root@host:~# systemctl enable mariadb
```

2.7 NVIDIA driver

2.7.1 NVIDIA driver

Note that this section can be skipped if working on a virtual machine (GPU can not be used vor EasyWave on a VM).

1. Login as root. If installed, remove the NVIDIA driver from the distribution:

   ```
   root@host:~# apt purge nvidia*
   ```

2. Add the graphics drivers PPA:

   ```
   root@host:~# apt install dirmngr ca-certificates software-properties-common apt-transport-https dkms wget root@host:~# wget -O- https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/3bf863cc.pub | gpg –dearmor | tee /usr/share/keyrings/nvidia-drivers.gpg
   root@host:~# echo 'deb [signed-by=/usr/share/keyrings/nvidia-drivers.gpg] https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/
   /' | tee /etc/apt/sources.list.d/nvidia-drivers.list
   root@host:~# add-apt-repository contrib
   root@host:~# apt update
   root@host:~# apt upgrade
   ```

3. Install NVIDIA drivers with:

   ```
   root@host:~# apt search nvidia-driver-*
   root@host:~# apt install nvidia-driver
   ```

2.8 gsm

gempa software manager (GSM). GSM is a packet manager which supports to install SeisComP as well as gempa packages from a configured remote package repository. Database migration is not supported yet. However gsm informs the user about required migrations.

You can find a detailed description of the configuration options under `https://docs.gempa.de/gsm/README.html`.

2.8.0.1 Packages install

1. Login as root
2. Run the commands below to install the required libraries.

   ```
   root@host:~# apt install python3-pip python3-requests python3-cryptography python3-pytzdata python3-dateutil python3-tzlocal
   ```

2.8.1 Installation

1. Login as sysop
2. Change to your home directory and create a new directory:

```
sysop@host:~$ cd
sysop@host:~$ mkdir install
sysop@host:~$ cd install
```

3. Get the latest gsm package and saved it in /home/sysop/install.

```
sysop@host:~$ wget https://data.gempa.de/gsm/gempa-gsm.tar.gz
```

```
sysop@host:~$ tar xf gempa-gsm.tar.gz
sysop@host:~$ cd gsm
```

4. Run the commands below to install the required libraries via pip.

```
sysop@host:~$ python3 -m pip install --user humanize natsort tqdm configparser
```

5. Run gsm setup

```
sysop@host:~$ ./gsm setup
```

or

```
sysop@host:~$ vim gsm.conf
```

Add the following lines and adapt user, password and base_url to your gempa account:

```
[Settings]
os = rhel
os_version = 8
arch = x86_64
base_url = https://data.gempa.de/packages/xxx
user = xxx
password = xxx
release = 5
download_dir = /home/sysop/install/gsm/packages
install_path = /home/sysop/seiscomp
data_dir = /home/data
verify = True
debug = False
sync_dir = /home/sysop/install/gsm/sync
local_dir = /home/sysop/install/gsm/local
license_info_days = 30
```

6. Get latest package list

```
sysop@host:~$ ./gsm update
```

Chapter 3

SeisComP and gempa Modules

3.1 SeisComP

SeisComP

You can find a detailed description of the configuration options under `https://docs.gempa.de/seiscomp/current/`.

3.1.0.1 Packages install

1. Login as root
2. Install base dependencies via apt

```
root@host:~# apt install libxml2 libboost-filesystem1.74.0 libboost-iostreams1.74.0 libboost-thread1.74.0
libboost-program-options1.74.0 libboost-regex1.74.0 libboost-system1.74.0 libssl3 libncurses6 libmysqlclient21
libmariadb3 libpq5 libpython3.10 python3-numpy
```

3. Install mariadb dependencies via apt

```
root@host:~# apt install mariadb-server mariadb-client
```

4. Install gui dependencies via apt

```
root@host:~# apt install libqt5gui5 libqt5xml5 libqt5opengl5 libqt5sql5-sqlite libqt5svg5
```

3.1.0.2 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm
sysop@host:~$ ./gsm install seiscomp
```

3.1.1 Initialize SeisComP Database MariaDB

1. Login as sysop

```
root@host:~# mysql -u root -p
```

```
CREATE DATABASE seiscomp CHARACTER SET utf8 COLLATE utf8_bin;
   grant usage on seiscomp.* to sysop@localhost identified by 'sysop';
   grant all privileges on seiscomp.* to sysop@localhost;
   grant usage on seiscomp.* to sysop@'%' identified by 'sysop';
   grant all privileges on seiscomp.* to sysop@'%';
   flush privileges;
   quit
```

2. Create the SeisComP database tables

```
sysop@host:~$ mysql -u sysop  -p   seiscomp  < /home/sysop/seiscomp/share/db/mysql.sql
```

3.1.2 Initial SeisComP environment

1. Login as sysop
2. Add SeisComP environment variables:

```
sysop@host:~$ /home/sysop/seiscomp/bin/seiscomp print env >> /home/sysop/.bashrc
sysop@host:~$ source /home/sysop/.bashrc
```

3.1.3 Initial SeisComP setup

Alternatively the initial setup can be also done through the setup wizard of scconfig.

1. Login as sysop
2. Run the command below from the command line and follow the instructions:

```
sysop@host:~$ seiscomp setup
```

```
===============================================================
SeisComP setup
===============================================================
This initializes the configuration of your installation.
If you already made adjustments to the configuration files
be warned that this setup will overwrite existing parameters
with default values. This is not a configurator for all
options of your setup but helps to setup initial standard values.
---------------------------------------------------------------------Hint: Entered values starting with a dot (.) are handled
as commands. Available commands are:
quit: Quit setup without modification to your configuration.
back: Go back to the previous parameter.
help: Show help about the current parameter (if available).
If you need to enter a value with a leading dot, escape it
with backslash, e.g. "\.value".
---------------------------------------------------------------------
Agency ID []: XXX
Datacenter ID []: XXX
Organization string []: XXX
Enable database storage [yes]:
0) mysql
MySQL server.
```

```
1) postgresql
Postgresql server. There is currently no support in setup to create the
database for you. You have to setup the database and user accounts on
your own. The database schema is installed under share/db/postgresql.sql.
Note that the database encoding should be UTF8 and that you need
to set the encoding to 'escape' for PostgreSQL >= 9,
e.g. "ALTER DATABASE seiscomp SET bytea_output TO 'escape';"
Database backend [0]:
Create database [yes]:
MYSQL root password (input not echoed) []:
Drop existing database [no]: yes
Database name [seiscomp]:
Database hostname [localhost]:
Database read-write user [sysop]:
Database read-write password [sysop]:
Database public hostname [localhost]: backup
Database read-only user [sysop]:
Database read-only password [sysop]:
Finished setup
--------------
P) Proceed to apply configuration
B) Back to last parameter
Q) Quit without changes
Command? [P]:
Running setup
-------------
* setup kernel
* setup scmaster
+ Create MYSQL database
+ Found MYSQL server version 5.5.31-log
+ Drop database seiscomp
+ Create database seiscomp
+ Setup user roles
+ Create tables
* setup trunk
* setup trunk
```

3.1.4 Autostart systemd

1. Login as root user and change into the system directory of the systemd configuration with:

```
root@host:~# cd /etc/systemd/system/
```

2. Run the commands below to install the required libraries.
   Create a new service named seiscomp and add the following lines:

```
root@host:~# vim /etc/systemd/system/seiscomp.service
```

```
[Unit]
Description=SeisComP
After=syslog.target network.target mariadb.service

[Service]
Type=oneshot
RemainAfterExit=yes
User=sysop
Group=sysop
ExecStart=/home/sysop/seiscomp/bin/seiscomp start
ExecStop=/home/sysop/seiscomp/bin/seiscomp stop

# Environment file containing key=value pairs. Must not contain bash syntax.
# Note: The same file may be sourced in your ~/.bashrc via
# export $(cat /home/sysop/seiscomp/etc/env/proc1 | xargs)
#EnvironmentFile=/home/sysop/seiscomp/etc/env/proc1

# depends on the system configuration, use same value as in /etc/security/limits.conf
```

```
LimitNOFILE=10000

[Install]
WantedBy=multi-user.target
```

3. Enable the service at system start with:

```
root@host:~# systemctl enable seiscomp
```

4. To start the service run:

```
root@host:~# systemctl start seiscomp
```

3.1.4.1 SELinux

3.1.4.1.1 SELinux status information Display the status of the SELinux extension with:

1. Login as root

```
root@host:~# sestatus
```

3.1.4.1.2 Disable SELinux To permanently disable SELinux on RHEL run

1. Set the disabled mode

```
sed -i 's/enforcing/disabled/g' /etc/selinux/config
```

2. Restart the system to apply the changes

```
root@host:~# reboot
```

We do not recommend to disable the SELinux extension permanently but it might be useful for testing.

3.1.4.1.3 Create new SELinux rules for seiscomp autostart via systemd

1. Login as root
2. To install the SElinux tools package, enter this

```
root@host:~# apt install policycoreutils-python audit
```

3. Run the audit2allow -w -a command to produce a human-readable description of why access was denied.

```
root@host:~# cat /var/log/audit/audit.log | grep seiscomp | audit2allow -M seiscomp
```

```
root@host:~# semodule -i seiscomp.pp
```

4. Start seiscomp

```
root@host:~# systemctl start seiscomp
```

3.1.5 Crontab

1. Login as sysop:

```
sysop@host:~$ crontab -e
```

2. add the following line

```
*/3 * * * * /home/sysop/seiscomp/bin/seiscomp check >/dev/null 2>&1
```

3.1.6 Version control

The SeisComP directories of all systems are under version control to be able to track configuration changes. We use Git as version control system cause it is simple and powerful. For more information about Git see `https://git-scm.com/`. To ensure availibilty when one host is down, we recommend to use two remote repositories. This subsection describes how to start from scratch to put SeisComP installation under version control. Every machine has its own branch to track the changes. Global changes must be first applied to the master and finally merged into branches.

3.1.6.1 git aliases

Login as sysopand create the file /home/sysop/.bash_aliases with an text editor and add the following lines:

```
alias git-uo='git -c "user.name=User One" -c "user.email=user_one@whatever.de"'
alias git-ut='git -c "user.name=User Two" -c "user.email=user_two@whatever.de"'
```

To actived the aliases for the current shell use

```
sysop@host:~$ source /home/sysop/.bash_aliases
```

By default RHEL based systems will not source this alias file. It is recommended to source the file in your `/home/sysop/.bashrc`:

```
sysop@host:~$ echo source "/home/sysop/.bash_aliases" >> /home/sysop/.bashrc
```

3.1.6.2 Remote repository

1. Login as root
2. Create a new directory and change the ownership to sysop:

```
root@host:~# mkdir -p /home/data/git/seiscomp.git
root@host:~# chown -R sysop:sysop /home/data/git/seiscomp.git
```

3. Login as sysopuser
4. Change the directory

```
sysop@host:~$ cd /home/data/git/seiscomp.git
```

5. To initialize the remote repository, enter this:

```
sysop@host:~$ git --bare init --shared=group
```

3.1.6.3 Local repository

To track only certain changes in the SeisComP directory a special configuration file must be created to ignore temporary files.

1. Login as sysop
2. Change the directory

   ```
   sysop@host:~$ cd /home/sysop/seiscomp
   ```

3. Create the file .gitignore with an text editor and add the following lines:

   ```
   *.pyc
   var/
   *.mbtiles
   ```

4. Initialize the local repository with:

   ```
   sysop@host:~$ git init
   ```

   Run the commands below to track the changes of the SeisComP directories:

   ```
   sysop@host:~$ git add .gitignore bin/ include/ lib/ sbin/ share/ etc/
   ```

5. Create bash alias to ease committing with correct user information. For each user add a line in ~/.bash_aliases, e.g.

   ```
   alias commit-sh='git commit --author="Ralph Henneberger <ralix@gempa.de>"'
   ```

6. Enter the following commands to setup the remote repositories:

   ```
   sysop@host:~$ git remote add origin ssh://sysop@proc1/home/data/git/seiscomp.git
   ```

7. (Optional) add second or third push remote for redundancy

   ```
   sysop@host:~$ git remote set-url --add --push origin ssh://username@proc2/home/data/git/seiscomp.git
   ```

8. Run the command below and check the output carefully:

   ```
   sysop@host:~$ git status
   ```

9. Commit the changes with:

   ```
   sysop@host:~$ git-?? commit -a
   ```

10. To update the remote repositories with Git 1.6.3 or newer:

    ```
    sysop@host:~$ git push --set-upstream origin master
    ```

    or

    ```
    sysop@host:~$ git push origin master
    ```

3.1.6.4 Redundant Systems

Redundant systems typically share a similar configuration and deviate only in a few configuration variables. Instead of maintaining two branches, which need to be merged to keep configuration changes in sync, it is far easier to use only one branch and manage small configuration differences in environment files. These files are

- stored under `/home/sysop/seiscomp/etc/env`
- named, e.g., `proc1` and `proc2` in case of a processing system
- define environment variables as key-value pairs which are used in SeisComP configuration files
- put under version control

The content of such an file might look like:

```
SC_DB_READ='sysop:\user@proc1/seiscomp'
SC_AGENCY_ID=AGENCY1
SC_QL2SC_SYNC_EVENT_ATTRIBUTES=false
SC_PEER_HOST=proc2
SC_GDS_ID=gds1
SC_GDS_SLAVE=false
```

To load the correct file on each system one must

1. Define the `EnvironmentFile` parameter in the SeisComP systemd service unit under `/etc/systemd/system/seiscomp.service`
2. Export the environment variables in `/home/sysop/.bashrc`:

   ```
   export $(cat /home/sysop/seiscomp/etc/env/proc1 | xargs)
   ```

3.2 TOAST Client

TOAST (Tsunami Observation And Simulation Terminal) is a software for tsunami simulation and verification giving a quick hazard assessment. The results can be verified by oceanographic sensors such as tide gauges or buoys. TOAST is developed by gempa GmbH, a spin-off from GFZ Potsdam (developer of the real-time earthquake processing and analysis system SeisComP). gempa is the SeisComP service and development company and has contributed to key components of the German Indonesian Tsunami Early Warning System (GITEWS).

During the development of the GITEWS software used by BMKG in Jakarta it soon became apparent that the system was complex and specialized to an extent which made it hard to offer to other institutes. Therefore gempa started the development of TOAST, a very flexible tsunami early warning software with high scalability.

TOAST is perfectly complementing SeisComP for the implementation of a fully functional tsunami warning system. TOAST is fully optimized for its application: Tsunami Early Warning. While conventional tsunami early warning systems are based on huge databases of pre-calculated scenarios. By default TOAST uses an on-the-fly simulation approach. Because of this approach TOAST can react to any atypical events, for example earthquakes in unconsidered areas or earthquakes with atypical rupture mechanisms. Additional to this on-the-fly simulation, TOAST's flexible simulation interface also allows integration of existing pre-calculated scenario databases.

You can find a detailed description of the configuration options under `https://docs.gempa.de/toast-client/current/`.

3.2.1 Packages install

1. Login as root
2. Install dependencies via apt

```
root@host:~# apt install qt5webkit5
```

3. Install mencoder via apt (optional)

```
root@host:~# apt install menocder
```

3.2.1.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm
sysop@host:~$ ./gsm install toast-client
```

3.2.1.2 Add Data

3.2.1.2.1 Forecast Zones and Points

1. Login as root

```
root@host:~#  mkdir -p /home/data/forecastzones
root@host:~#  chown -R sysop:sysop  /home/data/forecastzones
```

2. Login as sysop
3. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm
sysop@host:~$ ./gsm install forecastzones-rtsp
```

3.2.1.2.2 Scripts

```
sysop@host:~$ cd /home/sysop/seiscomp/share/toast/scripts
sysop@host:~$ cp disseminate.sh.example disseminate.sh
sysop@host:~$ cp makevideo.sh.example makevideo.sh
```

3.2.1.3 Configuration

3.2.1.3.1 Add Fault Information

This subsection describes the installation and configuration of fault information. The gss package comes with a world wide set of fault lines for all major earthquake zones. This information is used by the TEWS (Tsunami Early Warning System) system to generate rupture areas based on earthquake parameters.

```
sysop@host:~$ cd /home/sysop/seiscomp/share/toast/
sysop@host:~$ cp faults.xml.example faults.xml
```

3.2.2 Configuration Files

3.2.2.1 toast.cfg

Open/Create configuration file and add following lines:

```
sysop@host:~$ vim /home/sysop/seiscomp/etc/toast.cfg
```

```
# SeisComP applications access waveform data through the RecordStream
# interface. Read the SeisComP documentation for a list of supported services
# and their configuration.
# This parameter configures RecordStream source URL.
# Format: [service:///]location[#type]
# "service": The name of the RecordStream implementation. If not given,
# "file://" is implied.
# IMPORTANT: This parameter replaces former RecordStream configurations in
# "recordstream.service" and "recordstream.source" which have been removed.
recordstream = caps://localhost:18002

# Define the URL of the messaging server to connect to. URL format is
# [scheme:///]host[:port][/queue]. Default scheme: scmp, default port: 18180,
# default queue: production. Examples:
# All default: localhost
# Specific host: scmp://10.0.1.4:18180/production
# Specific host and SSL encryption: scmps://10.0.1.4:18180/production
connection.server = localhost:18181/production

# Show cities defined in "citiesXML" on maps
scheme.map.showCities = false

# Registration of forecast zones.
forecastZones = rtsp_2018

# Absolute filename without file extension
forecastZones.rtsp_2018.filename = /home/data/forecastzones/rtsp/6.3/cfz/CFZ_Version_2018Mar14

# Registration of live tabs.
liveTabs = ptwc

# Title of the tab
liveTab.ptwc.title = PTWC

# Template tree entry point, e.g., /National. This option can be used to limit
# the template selection of the live tab to a specific subtree.
liveTab.ptwc.entryPoint = /PTWC
```

3.3 Misc

3.3.1 Maps

3.3.1.1 Installation

Most GUI and some command line SeisComP applications require an extra map package which must be installed and configured. The system supports different map types one of them is MBTiles. You can find a detailed description of the configuration options under `https://docs.gempa.de/plugins/current/base/global_mbtiles.html`.

1. Login as root and create a data directory if it does not exist

```
root@host:~# mkdir -p /home/data/maps
root@host:~# chown -R sysop:sysop  /home/data/maps
```

2. Login as sysop
3. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm
sysop@host:~$ ./gsm install mapprojections world-11
```

3.3.1.2 Configuration Files

Note: Just choose only one of the following configurations for your maps.

3.3.1.2.1 global.cfg

3.3.1.2.1.1 Gempa

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp/etc/global.cfg
```

```
plugins = ${plugins},mbtiles,mapprojections

# conversion specification which starts with is introduced by the character %
# followed by a conversion specifier. Valid specifiers are s (replaced by tile
# ID), l (tile level), c (tile column) and r (tile row). An example for using
# the OpenStreetMap file structure is /path/to/maps/%l/%c/%r.png.

map.location = /home/data/maps/world-11/world-11.mbtiles

# Used to distinguish tile store implementations provided by plug-ins.
map.type = mbtiles

# Restricts the zoom level to discrete steps matching the resolution of the map
# tiles. The intention is to avoid interpolation when operating with tiles sets
# which use small font sizes.
map.mercator.discrete = true

# Zoom sensitivity of the map
map.zoom.sensitivity = 1

# Apply bilinear filtering to maps. The bilinear filter improves the visual
# quality but decreases performance slightly. It is only used for static map
# images. Not while dragging.
scheme.map.bilinearFilter = false

# Show cities defined in "citiesXML" on maps
scheme.map.showCities = false

# SeisComP ships with the rectangular projection built-in. Other projections
# may be provided through plugins.
scheme.map.projection = Mercator

# The precision of depth values.
scheme.precision.depth = 1

# The precision of lat/lon values.
scheme.precision.location = 3

# Precision of uncertainty values, e.g. latitude errors.
scheme.precision.uncertainties = 1

# The color grid line color of the map.
scheme.colors.map.grid = ff0000ff
```

3.3.1.2.1.2 OpenStreetMap

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp/etc/global.cfg
```

```
plugins = ${plugins},mbtiles,mapprojections

# conversion specification which starts with is introduced by the character %
# followed by a conversion specifier. Valid specifiers are s (replaced by tile
# ID), l (tile level), c (tile column) and r (tile row). An example for using
# the OpenStreetMap file structure is /path/to/maps/%l/%c/%r.png.

map.location = https://tile.openstreetmap.org/\l/%c/%r.png

# Used to distinguish tile store implementations provided by plug-ins.
map.type = osm

# Restricts the zoom level to discrete steps matching the resolution of the map
# tiles. The intention is to avoid interpolation when operating with tiles sets
# which use small font sizes.
map.mercator.discrete = true

# Zoom sensitivity of the map
map.zoom.sensitivity = 1

# Apply bilinear filtering to maps. The bilinear filter improves the visual
# quality but decreases performance slightly. It is only used for static map
# images. Not while dragging.
scheme.map.bilinearFilter = false

# Show cities defined in "citiesXML" on maps
scheme.map.showCities = false

# SeisComP ships with the rectangular projection built-in. Other projections
# may be provided through plugins.
scheme.map.projection = Mercator

# The precision of depth values.
scheme.precision.depth = 1

# The precision of lat/lon values.
scheme.precision.location = 3

# Precision of uncertainty values, e.g. latitude errors.
scheme.precision.uncertainties = 1

# The color grid line color of the map.
scheme.colors.map.grid = ff0000ff
```

3.3.1.2.1.3 Google

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp/etc/global.cfg
```

```
plugins = ${plugins},mbtiles,mapprojections

# conversion specification which starts with is introduced by the character %
# followed by a conversion specifier. Valid specifiers are s (replaced by tile
# ID), l (tile level), c (tile column) and r (tile row). An example for using
# the OpenStreetMap file structure is /path/to/maps/%l/%c/%r.png.

map.location = map.location = http://mt1.google.com/vt/lyrs=y\&x=\c&y=%r&z=%l

# Used to distinguish tile store implementations provided by plug-ins.
map.type = osm
```

```
# Restricts the zoom level to discrete steps matching the resolution of the map
# tiles. The intention is to avoid interpolation when operating with tiles sets
# which use small font sizes.
map.mercator.discrete = true

# Zoom sensitivity of the map
map.zoom.sensitivity = 1

# Apply bilinear filtering to maps. The bilinear filter improves the visual
# quality but decreases performance slightly. It is only used for static map
# images. Not while dragging.
scheme.map.bilinearFilter = false

# Show cities defined in "citiesXML" on maps
scheme.map.showCities = false

# SeisComP ships with the rectangular projection built-in. Other projections
# may be provided through plugins.
scheme.map.projection = Mercator

# The precision of depth values.
scheme.precision.depth = 1

# The precision of lat/lon values.
scheme.precision.location = 3

# Precision of uncertainty values, e.g. latitude errors.
scheme.precision.uncertainties = 1

# The color grid line color of the map.
scheme.colors.map.grid = ff0000ff
```

3.3.1.2.1.4 ArcGIS

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp/etc/global.cfg
```

```
plugins = ${plugins},mbtiles,mapprojections

# conversion specification which starts with is introduced by the character %
# followed by a conversion specifier. Valid specifiers are s (replaced by tile
# ID), l (tile level), c (tile column) and r (tile row). An example for using
# the OpenStreetMap file structure is /path/to/maps/%l/%c/%r.png.

map.location = map.location = https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/\l/%r/%c.jpeg

# Used to distinguish tile store implementations provided by plug-ins.
map.type = osm

# Restricts the zoom level to discrete steps matching the resolution of the map
# tiles. The intention is to avoid interpolation when operating with tiles sets
# which use small font sizes.
map.mercator.discrete = true

# Zoom sensitivity of the map
map.zoom.sensitivity = 1

# Apply bilinear filtering to maps. The bilinear filter improves the visual
# quality but decreases performance slightly. It is only used for static map
# images. Not while dragging.
scheme.map.bilinearFilter = false

# Show cities defined in "citiesXML" on maps
scheme.map.showCities = false

# SeisComP ships with the rectangular projection built-in. Other projections
# may be provided through plugins.
scheme.map.projection = Mercator
```

```
# The precision of depth values.
scheme.precision.depth = 1

# The precision of lat/lon values.
scheme.precision.location = 3

# Precision of uncertainty values, e.g. latitude errors.
scheme.precision.uncertainties = 1

# The color grid line color of the map.
scheme.colors.map.grid = ff0000ff
```

Chapter 4

SeisComP (TEWS) and gempa Modules (TEWS)

The following instructions describe the installation of a TEWS (Tsunami Early Warning System). which receives earthquake information, such as magnitude and location, from a separate location and from a separate SeisComP system. We recommend to separate both systems , i.e. each system has its own master for the messaging connection.

This has the following advantages:

* Inventories are easier to manage
* The existing SeisComP configuration does not have to be adapted
* Errors do not affect the operation of the SeisComP system

4.1 gsm

gempa software manager (GSM). GSM is a packet manager which supports to install SeisComP as well as gempa packages from a configured remote package repository. Database migration is not supported yet. However gsm informs the user about required migrations.

You can find a detailed description of the configuration options under `https://docs.gempa.de/gsm/README.html`.

4.1.0.1 Packages install

1. Login as root
2. Run the commands below to install the required libraries.

```
root@host:~# apt install python3-pip python3-requests python3-cryptography python3-pytzdata python3-dateutil python3-tzlocal
```

4.1.1 Installation

1. Login as sysop
2. Change to your home directory and create a new directory:

```
sysop@host:~$ cd
sysop@host:~$ mkdir install
sysop@host:~$ cd install
```

3. Get the latest gsm package and saved it in /home/sysop/install.

```
sysop@host:~$ wget https://data.gempa.de/gsm/gempa-gsm.tar.gz
```

```
sysop@host:~$ mkdir /home/sysop/install/gsm-tews
sysop@host:~$ tar xf gempa-gsm.tar.gz -C /home/sysop/install/gsm-tews  --strip-components=1
sysop@host:~$ cd /home/sysop/install/gsm-tews
```

4. Run the commands below to install the required libraries via pip.

```
sysop@host:~$ python3 -m pip install --user humanize natsort tqdm configparser
```

5. Run gsm setup

```
sysop@host:~$ ./gsm setup
```

or

```
sysop@host:~$ vim gsm.conf
```

Add the following lines and adapt user, password and base_url to your gempa account:

```
[Settings]
os = rhel
os_version = 8
arch = x86_64
base_url = https://data.gempa.de/packages/xxx
user = xxx
password = xxx
release = 5
download_dir = /home/sysop/install/gsm-tews/packages
install_path = /home/sysop/seiscomp-tews
data_dir = /home/data
verify = True
debug = False
sync_dir = /home/sysop/install/gsm-tews/sync
local_dir = /home/sysop/install/gsm-tews/local
license_info_days = 30
```

6. Get latest package list

```
sysop@host:~$ ./gsm update
```

4.2 SeisComP

SeisComP

You can find a detailed description of the configuration options under `https://docs.gempa.de/seiscomp/current/`.

4.2.0.1 Packages install

1. Login as root
2. Install base dependencies via apt

```
root@host:~# apt install libxml2 libboost-filesystem1.74.0 libboost-iostreams1.74.0 libboost-thread1.74.0 libboost-program-options1.74.0 libboost-regex1.74.0 libboost-system1.74.0 libssl3 libncurses6 libmysqlclient21 libmariadb3 libpq5 libpython3.10 python3-numpy
```

3. Install mariadb dependencies via apt

```
root@host:~# apt install mariadb-server mariadb-client
```

4. Install gui dependencies via apt

```
root@host:~# apt install libqt5gui5 libqt5xml5 libqt5opengl5 libqt5sql5-sqlite libqt5svg5
```

4.2.0.2 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install seiscomp
```

4.2.1 Configuration Files

4.2.1.1 global.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/global.cfg
```

```
# Sets the datacenter ID which is primarily used by Arclink and its tools.
# Should not contain spaces.
datacenterID = XXX

# Defines the agency ID used to set creationInfo.agencyID in data model
# objects. Should not contain spaces.
agencyID = XXX

# Organization name used mainly by ArcLink and SeedLink.
organization = XXX

# Defines a list of core modules loaded at startup.
core.plugins = dbmysql
```

4.2.2 Initialize SeisComP Database MariaDB

1. Login as sysop

```
root@host:~# mysql -u root -p
```

```
CREATE DATABASE seiscomp-tews CHARACTER SET utf8 COLLATE utf8_bin;
   grant usage on seiscomp-tews.* to sysop@localhost identified by 'sysop';
   grant all privileges on seiscomp-tews.* to sysop@localhost;
   grant usage on seiscomp-tews.* to sysop@'%' identified by 'sysop';
   grant all privileges on seiscomp-tews.* to sysop@'%';
   flush privileges;
   quit
```

2. Create the SeisComP database tables

```
sysop@host:~$ mysql -u sysop  -p seiscomp-tews  < /home/sysop/seiscomp-tews/share/db/mysql.sql
```

4.2.3 Create SeisComP aliase

1. Login as sysop:
2. Create the file /home/sysop/.bash_aliases with an text editor and add the following lines:

```
alias sc-tews='/home/sysop/seiscomp-tews/bin/seiscomp'
```

3. Read and actived the aliases:

```
sysop@host:~$ source /home/sysop/.bash_aliases
```

4.2.4 Autostart systemd

1. Login as root user and change into the system directory of the systemd configuration with:

```
root@host:~# cd /etc/systemd/system/
```

2. Run the commands below to install the required libraries.
   Create a new service named seiscomp and add the following lines:

```
root@host:~# vim /etc/systemd/system/seiscomp-tews.service
```

```
[Unit]
Description=SeisComP_TEWS
After=syslog.target network.target postgresql.service

[Service]
Type=oneshot
RemainAfterExit=yes
User=sysop
Group=sysop
ExecStart=/home/sysop/seiscomp-tews/bin/seiscomp start
ExecStop=/home/sysop/seiscomp-tews/bin/seiscomp stop

# Environment file containing key=value pairs. Must not contain bash syntax.
# Note: The same file may be sourced in your ~/.bashrc via
# export $(cat /home/sysop/seiscomp/etc/env/proc1 | xargs)
#EnvironmentFile=/home/sysop/seiscomp/etc/env/proc1

# depends on the system configuration, use same value as in /etc/security/limits.conf
LimitNOFILE=30000:30000

[Install]
WantedBy=multi-user.target
```

3. Enable the service at system start with:

```
root@host:~# systemctl enable seiscomp-tews
```

4. To start the service run:

```
root@host:~# systemctl start seiscomp-tews
```

4.2.4.1 SELinux

4.2.4.1.1 SELinux status information Display the status of the SELinux extension with:

1. Login as root

```
root@host:~# sestatus
```

4.2.4.1.2 Disable SELinux To permanently disable SELinux on RHEL run

1. Set the disabled mode

```
sed -i 's/enforcing/disabled/g' /etc/selinux/config
```

2. Restart the system to apply the changes

```
root@host:~# reboot
```

We do not recommend to disable the SELinux extension permanently but it might be useful for testing.

4.2.4.1.3 Create new SELinux rules for seiscomp-tews autostart via systemd

1. Login as root
2. To install the SElinux tools package, enter this

```
root@host:~# apt install policycoreutils-python audit
```

3. Run the audit2allow -w -a command to produce a human-readable description of why access was denied.

```
root@host:~# cat /var/log/audit/audit.log | grep seiscomp | audit2allow -M seiscomp
```

```
root@host:~# semodule -i seiscomp.pp
```

4. Start seiscomp-tews

```
root@host:~# systemctl start seiscomp-tews
```

4.2.5 Crontab

1. Login as sysop:

```
sysop@host:~$ crontab -e
```

2. add the following line

```
*/3 * * * * /home/sysop/seiscomp-tews/bin/seiscomp check >/dev/null 2>&1
```

4.3 Acquisition

4.3.1 CAPS

The Common Acquisition Protocol Server (CAPS) was developed to fulfill the needs to transfer multi-sensor data from the station to the data center. As nowadays more and more stations with co-located sensors like broadband seismometer, accelerometer, GPS, temperature, video cameras, etc. are build up, a acquisition protocol is required, which can efficiently handle low- and high-sampled data through one unified protocol. The data conversion is realized by plug-ins which send data directly to CAPS. There is no plug-in registration at server side which simplifies the configuration. The basic configuration steps are explained in this section.

You can find a detailed description of the configuration options under `https://docs.gempa.de/caps/`.

4.3.1.1 Maximum open files

By default the number of open files is limited by the operating system to a certain number, e.g., 1024. Some applications require more open files to run. This section describes how to change this value if required. Please use a value that is sufficient for all applications and still provides some safety margin.

4.3.1.1.1 Configuration

1. Login as root
2. Open the file */etc/security/limits.conf* and add the following lines to increase the maximum number of open files to, e.g., 30000

```
sysop  soft nofile 30000
```

```
sysop  hard nofile 30000
```

3. Apply the new configuration

```
root@host:~# sysctl -p /etc/sysctl.conf
```

4.3.1.2 Installation

1. Login as root and create a data directory if it does not exist

```
root@host:~# mkdir -p /home/data/archive/caps
root@host:~# chown -R sysop:sysop /home/data/archive/caps
```

2. Login as sysop
3. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install caps-server caps-plugins caps-tools
```

4.3.1.3 Configuration Files

4.3.1.3.1 caps.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/caps.cfg
```

```
# Defines the path to the archive directory.
AS.filebase = /home/data/archive/caps/archive

# Defines the server port for HTTP connections. By default the Web interface is
# disabled.
AS.http.port = 18081

# Preallocation size of data files.
AS.filebase.preallocationSize = 0

# The maximum number of open files. Because a stream file can have an
# associated index file this value is half of the physically opened files in
# worst case.
AS.filebase.cache.openFileLimit = 5000

# Limit of cached files in total. This value affects also files that are
# actually explicitly closed by the application. CAPS will keep them open
# (respecting the openFileLimit parameter) as long as possible and preserve a
# file handle to speed up reopening the file later.
AS.filebase.cache.unusedFileLimit = 5000

# Defines the number of days to keep data. Format is: stream id : days. The
# default is to keep data forever.
AS.filebase.keep = VZ.*.*.*:-1,*.*.*.*:30
```

2. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable caps
sysop@host:~$ sc-tews  start caps
```

4.3.1.4 Test

1. Try to establish a connection to check if the server is up and running

```
sysop@host:~$ telnet localhost 18002
```

Type **hello** and check that the terminal output is similar to

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
hello
4CAPS (gempa GmbH) v2023.128#aeb4e5f5b
API=5
WELCOME
bye
```

4.3.1.5 IOC data import

This section describes the IOC tide gauge data integration into the TEWS system.

4.3.1.5.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install tidegauges-ioc recordstream
```

4.3.1.5.2 Optional

1. **Optional** extract VZ data for the last major earthquakes since 2007:

```
sysop@host:~$ mkdir -p /home/data/archive/caps/
sysop@host:~$ cp archive.caps.mseed.ioc.vz.tar.gz /home/data/archive/caps
sysop@host:~$ cd /home/data/archive/caps/
sysop@host:~$ tar xf archive.caps.mseed.ioc.vz.tar.gz
```

4.3.1.5.3 Configuration Files

4.3.1.5.3.1 rs2caps.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/rs2caps.cfg
```

```
core.plugins = dbpostgresql

# Defines a list of modules loaded at startup.
plugins = ${plugins},rsioc

recordstream = ioc://ioc-sealevelmonitoring.org/service.php?maxThreads=2&interval=120

# File to store stream states
journal.file = @ROOTDIR@/var/run/rs2caps/journal

# Inventory XML to read the streams to be added from
database.inventory = @ROOTDIR@/etc/inventory/VZ.xml

ioc.sensorMap = @ROOTDIR@/etc/rs2caps/sensormap.txt

# Start time of data time window, default 'GMT'
streams.begin = "2023-09-04 00:00:00"
```

4.3.1.5.3.2 sensormap.txt

1. Create a new directory:

```
sysop@host:~$ mkdir -p /home/sysop/seiscomp-tews/etc/rs2caps
```

2. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/rs2caps/sensormap.txt
```

```
# Sensor Loc Description
RAD,40,radar
RA2,42,2st radar

PRS,50,pressure
PR1,51,1st pressure
PR2,52,2st pressure
P1,53»··1st pressure
P2,54,2st pressure
PR,55,pressure

BUB,60,bubbler
BUB1,61,1st bubbler
BUB2,62,2st bubbler
BBB,63,bubbler

PWL,70,primary water level
FLT,73,float
FLO,74,float
WLS,76,water level sensor
BWL,79,backup water level

ENC,80
ENB,81
```

3. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable rs2caps
sysop@host:~$ sc-tews  start rs2caps
```

4.4 GSS Server

GSS You can find a detailed description of the configuration options under `https://docs.gempa.de/gss/current/`.

4.4.0.1 Packages install

1. Login as root
2. Install dependencies via apt

```
root@host:~# apt install libqt5core5a
```

4.4.0.2 Installation

1. Login as root and create a data directory if it does not exist

```
root@host:~# mkdir -p /home/data/sourceregions
root@host:~# chown -R sysop:sysop  /home/data/sourceregions
```

2. Login as sysop
3. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install gss-server sourceregions
```

4.4.0.3 Add Fault Information

This subsection describes the installation and configuration of fault information. The gss package comes with a world wide set of fault lines for all major earthquake zones. This information is used by the TEWS system to generate rupture areas based on earthquake parameters.

```
sysop@host:~$ cd /home/sysop/seiscomp-tews/share/gss/
sysop@host:~$ cp faults.xml.example faults.xml
```

4.4.1 Configuration Files

4.4.1.1 gss.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/gss.cfg
```

```
# List of plugins loaded at startup. Separate multiple names by comma. Add
# ${plugins} to consider all previously read values.
plugins = ${plugins}, simeasywave2

# Set the logging level between 1 and 4 where 1=ERROR, 2=WARNING, 3=INFO and
# 4=DEBUG.
logging.level = 4

# Defines the server port for client requests.
GSS.port = 19010

# Usually more than one grid file is avaibable for the same earthquake location
# and the automatic system does not know which one to choose. Use this option
# to define which bathymetry grid should used for automatic processing(Strg +
# N).
# Registration of source regions. Each entry maps one or more regions to a
# bathymetry file. The entries are evaluated in FIFO order. If none of the
# entries matches the algorithm tries to select a fallback grid.
GSS.sourceRegions = io_rtsp

# Port to listen for HTTP request. Recommended HTTP port is 19080.
GSS.http.port = 19080

# Path to HTTP media files
GSS.path.media = @DATADIR@/gss/media

# Path to BNA file.
GSS.sourceRegion.io_rtsp.aoi = /home/data/sourceregions/indian_ocean_rtsp.bna

# Name of the bathymetry file without the extension
GSS.sourceRegion.io_rtsp.name = indian_ocean_rtsp

GSS.database=postgresql://sysop:sysop@localhost/gss
```

4.4.2 Initialize gss Database MariaDB

1. Login as sysop

```
root@host:~# mysql -u root -p
```

```
CREATE DATABASE seiscomp-tews CHARACTER SET utf8 COLLATE utf8_bin;
   grant usage on seiscomp-tews.* to sysop@localhost identified by 'sysop';
   grant all privileges on seiscomp-tews.* to sysop@localhost;
   grant usage on seiscomp-tews.* to sysop@'%' identified by 'sysop';
   grant all privileges on seiscomp-tews.* to sysop@'%';
   flush privileges;
   quit
```

2. Create the SeisComP database tables

```
sysop@host:~$ mysql -u sysop  -p gss  < /home/sysop/seiscomp-tews/share/db/gss/mysql.sql
```

4.4.3 Plugins

4.4.3.1 gss-plugins-easywave2

EasyWave is a simulation tool written in C++ to simulate tsunami generation and propagation. Due to its speed it is espacially suited for application in tsunami early warning. As EasyWave is not part of GSS, and rights owner is Andrey Babeyko (GFZ German Research Center for Geosciences, Potsdam, babeyko@gfz-potsdam.de), GSS is delivered only with a binary file of EasyWave.

You can find a detailed description of the configuration options at:
`https://docs.gempa.de/gss-plugin-easywave2/current/easywave2.html`.

4.4.3.1.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install gss-plugin-easywave2
```

4.4.3.1.2 Forecast Zones and Points

1. Login as root

```
root@host:~#  mkdir -p /home/data/forecastzones
root@host:~#  chown -R sysop:sysop  /home/data/forecastzones
```

2. Login as sysop
3. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install forecastzones-rtsp
```

4.4.3.1.3 Bathymetry Files (Grids) for EasyWave2

1. Login as root

```
root@host:~#  mkdir -p /home/data/bathymetry
root@host:~#  chown -R sysop:sysop  /home/data/bathymetry
```

2. Login as sysop
3. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install bathymetry
```

4.4.3.1.4 Configuration Files

Note: If a virtual machine is used here and the GPU is not available in the virtual machine, the following parameter must be set to false.

```
easywave2.gpu = false
```

4.4.3.1.4.1 easywave2.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/easywave2.cfg
```

```
# Set the logging level between 1 and 4 where 1=ERROR, 2=WARNING, 3=INFO and
# 4=DEBUG.
logging.level = 3
# Output interval of the arrival times grids in minutes.
easywave2.arrivalTimesStep = 30
# Directory for input bathymetry grid files. All files found directly in this
# folder and ending on '.grd' are interpreted as bathymetry files. The first
# matching grid (automatic calculation) or the file selected (interactive
# simulation) is passed to EasyWave.
easywave2.data = /home/data/bathymetry
# Registration of forecast points.
easywave2.forecastPoints = rtsp_2018
# Use GPU for processing if supported
easywave2.gpu = true
# Maximum time span for which the tsunami propagation is calculated.
easywave2.maxTime = 120
# Base directory of EasyWave outputs. The results of a particular simulation
# are expected in a subfolder named after the eventID specified in EasyWave
# input file.
easywave2.output = @CONFIGDIR@/gss/easywave2/output
# Absolute filename of the forecast point file in dBase format
easywave2.forecastPoints.rtsp_2018.filename = /home/data/forecastzones/rtsp/6.3/cfp/CFP_Version_2018Mar14.dbf
# List of alignments in strike direction in range [0, 1].
# 0.5: at hypocenter
# > 0.5: with strike
# < 0.5: against strike
# 0,0.5,1 creates simulations centered at the far ends and over the hypocenter
# of the original rupture.
easywave2.patches.strikeAlign = 0.25, 0.75
# Output interval of the ssh max grids in minutes.
easywave2.sshMaxStep = 30
```

2. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable gss
sysop@host:~$ sc-tews  start gss
```

4.4.3.2 gss-plugins-hysea

HySEA (Hyperbolic Systems and Efficient Algorithms) is a high-performance package developed by the EDANYA group at the University of Malaga, Spain, for the simulation of geophysical flows.

You can find a detailed description of the configuration options at:
`https://docs.gempa.de/gss-plugin-hysea/current/hysea.html`.

4.4.3.2.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install gss-plugin-hysea
```

4.4.3.2.1.1 Forecast Zones and Points

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install forecastzones-rtsp
```

4.4.3.2.2 Bathymetry Files (Grids) for Hysea

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install bathymetry
```

4.4.3.2.3 Configuration Files

4.4.3.2.3.1 hysea.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/hysea.cfg
```

```
hysea.maxTime = 60

hysea.grids=algeria, rtsp
hysea.grids.algeria.filename=/home/data/bathymetry/bat/algeria.bat
hysea.grids.algeria.width=1245
hysea.grids.algeria.height=480

hysea.grids.rtsp.filename=/home/data/bathymetry/bat/io_rtsp_3841_2761.bat
hysea.grids.rtsp.width=3841
hysea.grids.rtsp.height=2761
```

```
# Registration of forecast points.
hysea.forecastPoints = rtsp

# Absolute filename of the forecast point file in dBase format
hysea.forecastPoints.rtsp.filename = /home/data/forecastzones/rtsp/6.3/cfp/CFP_Version_2018Mar14.dbf
```

2. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable gss
sysop@host:~$ sc-tews  start gss
```

4.4.3.3 gss-plugins-geowarettt

Geowarettt

You can find a detailed description of the configuration options at:
`https://docs.gempa.de/gss-plugin-geowarettt/current/geowarett.html`

4.4.3.3.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install gss-plugin-geowarettt
```

4.4.3.3.2 Forecast Zones and Points

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install forecastzones-rtsp
```

4.4.3.3.3 Configuration Files

4.4.3.3.3.1 geowarettt.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/geowarettt.cfg
```

```
# Set the logging level between 1 and 4 where 1=ERROR, 2=WARNING, 3=INFO and
# 4=DEBUG.
logging.level = 3

# Geoware TTT executable.
geowarettt.binaryTTT = @DATADIR@/toast/ttt/ttt32

# Geoware TTT pick executable.
geowarettt.binaryTTTPick = @DATADIR@/toast/ttt/ttt_pick32
```

```
# Path to Geoware TTT bathymetry file in GMT i2 format(2 byte integer).
geowarettt.bathymetry = @DATADIR@/toast/ttt/ttt_topo_10m.i2

# Registration of forecast points.
geowarettt.forecastPoints = bom_sig, bom_fzp, rtsp_2018

# Absolute filename of an executable script file that is run after a simulation
# has been completed.
geowarettt.postProcessingScript = @DATADIR@/simgeowarettt/scripts/ttt_pp_example.sh

# Absolute filename of the forecast point file in dBase format.
geowarettt.forecastPoints.bom_fzp.filename = /home/data/forecastzones/bom/2.4/cfp/BOM_forecast_points.dbf

# Absolute filename of the forecast point file in dBase format.
geowarettt.forecastPoints.bom_sig.filename = /home/data/forecastzones/bom/2.4/cfp/locations_significant_fps.dbf

# Absolute filename of the forecast point file in dBase format.
geowarettt.forecastPoints.rtsp_2018.filename = /home/data/forecastzones/rtsp/6.3/cfp/CFP_Version_2018Mar14.dbf
```

2. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable gss
sysop@host:~$ sc-tews  start gss
```

4.4.3.4 gss-plugin-most

MOST

You can find a detailed description of the configuration options at:
`https://docs.gempa.de/gss-plugin-most/current/most.html`

4.4.3.4.1 Packages install

1. Login as root
2. Install dependencies via apt

```
root@host:~# apt install libnetcdf19
```

4.4.3.4.1.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install gss-plugin-most
```

4.4.3.4.2 Forecast Zones and Points

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install forecastzones-rtsp
```

4.4.3.4.3 Scenarios

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install scenarios-most
```

4.4.3.4.4 Configuration Files

4.4.3.4.4.1 bommost.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/bommost.cfg
```

```
# Set the logging level between 1 and 4 where 1=ERROR, 2=WARNING, 3=INFO and
# 4=DEBUG.
logging.level = 3

# Absolute path to the JSON scenario catalog file. This file contains the
# metadata information of the BOM scenario database.
bommost.catalog = /home/data/scenarios/most/t2_scenario_ruptures.json

# Registration of forecast points.
bommost.forecastPoints = bom, rtsp_2018

# Path to the SSH max files.
bommost.sshMaxData = /home/data/scenarios/most/grd_zips

# Path to the significant points (Australian National CFPs) result files. These
# are mapped to the Australian forecast zones.
bommost.cfpSigPointsData = /home/data/scenarios/most/travel_times

# Path to RTSP forecast zones result files. Note that here, forecast zones are
# mapped back to forecast points.
bommost.cfzRtspData = /home/data/scenarios/most/zone_data

# Path to DART buoy result files. Currently contain only summary results and no
# time series.
bommost.poiDartData = /home/data/scenarios/most/travel_times

# Distance mode: 'faults' (returns all scenarios with same minimal rupture
# element distance to epicenter) or 'centroid' (returns scenario with minimal
# centroid distance to epicenter, only 1 scenario returned even if several at
# identical distance).
bommost.matching.distMode = faults

# Absolute filename of the forecast point file in dBase format.
bommost.forecastPoints.bom.filename = /home/data/forecastzones/bom/2.4/cfp/locations_significant_fps.dbf

# Absolute filename of the forecast point file in dBase format.
bommost.forecastPoints.rtsp_2018.filename = /home/data/forecastzones/rtsp/6.3/cfp/CFP_Version_2018Mar14.dbf
```

2. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable gss
sysop@host:~$ sc-tews  start gss
```

4.4.3.5 gss-plugins-tsunawi

Tsundabi/Tsunawi toast plugin

4.4.3.5.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install gss-plugin-tsunawi
```

4.4.3.5.2 Configuration Files

4.4.3.5.2.1 tsunawi.cfg

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/tsunawi.cfg
```

```
# Base directory of TsunAWI outputs.
tsunawi.output = @CONFIGDIR@/tsunawi/output

# Specifies the TsunAWI URL under which simulation data is requested.
tsunawi.server = tsunawi.tews

# Registration of forecast zones.
tsunawi.forecastPoints = ntwc

# Absolute filename of the forecast point file in dBase format
tsunawi.forecastPoints.ntwc.filename = @CONFIGDIR@/forecast/points/cfp_tsunawi.dbf
```

2. **Enable module(s)** and then **start**

```
sysop@host:~$ sc-tews  enable gss
sysop@host:~$ sc-tews  start gss
```

4.5 TOAST Server

TOAST-server (Tsunami Observation And Simulation Terminal Server)

You can find a detailed description of the configuration options under `https://docs.gempa.de/toast-server/current/`.

4.5.0.1 Installation

1. Login as sysop
2. Install the package via gsm

```
sysop@host:~$ cd /home/sysop/install/gsm-tews
sysop@host:~$ ./gsm install toast-server
```

4.5.0.2 Configuration

4.5.0.2.1 Add Fault Information

This subsection describes the installation and configuration of fault information. The gss package comes with a world wide set of fault lines for all major earthquake zones. This information is used by the TEWS (Tsunami Early Warning System) system to generate rupture areas based on earthquake parameters.

```
sysop@host:~$ cd /home/sysop/seiscomp-tews/share/toast/
sysop@host:~$ cp /home/sysop/seiscomp-tews/share/gss/faults.xml.example faults.xml
```

4.5.0.2.2 Add Templates

```
sysop@host:~$ cd /home/sysop/seiscomp-tews/share/toastd/templates
sysop@host:~$ cp ptwc_info.html.example ptwc_info.html
sysop@host:~$ cp ptwc_ocean.html.example ptwc_ocean.html
sysop@host:~$ cp ptwc_region.html.example ptwc_region.html
sysop@host:~$ cp ptwc_local.html.example ptwc_local.html
```

4.5.1 Configuration Files

4.5.1.1 scmaster.cfg

Note: With this setting the toastd will connect to the production queue of host and import events into the TOAST database as incidents according to its configuration.

```
queues.production.processors.messages.toastd.source = proc:18180
```

1. **Open/Create configuration file and add following lines:**

```
sysop@host:~$ vim /home/sysop/seiscomp-tews/etc/scmaster.cfg
```

```
# List of plugins loaded at startup. Separate multiple names by comma. Add
# ${plugins} to consider all previously read values.
plugins = ${plugins}, dmtsunami

# Set the logging level between 1 and 4 where 1=ERROR, 2=WARNING, 3=INFO and
# 4=DEBUG.
logging.level = 2

# Define a whitelist of agencies that are allowed for processing. Separate
# items by comma.
processing.whitelist.agencies = IGN

# Define a list of core modules loaded at startup.
core.plugins = dbpostgresql

# Enable messaging queues defined as profile in queues. The profile names are
# the final queue names.
queues = production

# Local bind address and port of the messaging system. 0.0.0.0:18180 accepts
# connections from all clients, 127.0.0.1:18180 only from localhost.
interface.bind = 0.0.0.0:18181

# Define the list of message groups added to the queue. If unset, then the
# defaultGroups will be used. A queue will always add the default group
```

```
# "STATUS_GROUP". This parameter overrides defaultGroups.
queues.production.groups = CONFIG, INVENTORY, AMPLITUDE, PICK, TSUNAMI, GUI

# List of plugins required by this queue. This is just a convenience parameter
# to improve configurations readability. The plugins can also be added to the
# global list of module plugins.
# Example: dbstore
queues.production.plugins = dbstore, toastd

# Interface name. For now, use "dbstore"to use a database.
# Use empty for testing or playbacks without a database.
queues.production.processors.messages = dbstore, toastd

# Selected the database driver to use. Database drivers are available through
# plugins. The default plugin is dbmysql which supports the MYSQL database
# server. It is activated with the core.plugins parameter.
queues.production.processors.messages.dbstore.driver = postgresql

# Set the database read connection which is reported to clients that connect to
# this server. If a remote setup should be implemented, ensure that the
# hostname is reachable from the remote computer.
queues.production.processors.messages.dbstore.read = sysop:sysop@localhost/toast

# Set the database write connection which is private to scmaster. A separate
# write connection enables different permissions on the database level for
# scmaster and clients.
queues.production.processors.messages.dbstore.write = sysop:sysop@localhost/toast

# If enabled then the database connection as configured in 'read' is not being
# returned to the client but the URL "proxy://". This URL tells the client to
# open the database via the websocket proxy at the messaging address, e.g.
# http://localhost/production/db. The same hostname and queue must be used as
# for the initial messaging connection.
queues.production.processors.messages.dbstore.proxy = true

# Configure the SeisComP connection URL. If the string starts with a slash then
# it is taken as relative queue name of this instance.
queues.production.processors.messages.toastd.source = 0.0.0.0:18180

# Configure the source database URL to read tsunami related objects from. If
# not given then it will be retrieved when connecting to this queue if the
# dbstore plugin is configured.
queues.production.processors.messages.toastd.database = postgresql://sysop:sysop@localhost/toast

# The GSS request URL for the native protocol.
queues.production.processors.messages.toastd.gss.url = gss://localhost:19010

# The HTTP URL to browse products.
queues.production.processors.messages.toastd.gss.browserURL = https://localhost:19443

# Defines the ordered list of templates groups.
queues.production.processors.messages.toastd.bulletins.groups = ptwc

# Name of the group.
queues.production.processors.messages.toastd.bulletins.group.ptwc.name = "PTWC"

#Defines the ordered list of templates.
queues.production.processors.messages.toastd.bulletins.group.ptwc.templates = ptwc_info,\
                  ptwc_local,\
                  ptwc_ocean,\
                  ptwc_region

# ID of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_info.id = IDY0001

# Name of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_info.name = "Info"

# Path to template file.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_info.file = "@DATADIR@/toastd/templates/ptwc_info.html"

# ID of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_local.id = IDY0002

# Name of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_local.name = "Local"
```

```
# Path to template file.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_local.file = "@DATADIR@/toastd/templates/ptwc_local.html"

# ID of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_ocean.id = IDY0003

# Name of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_ocean.name = "Ocean"

# Path to template file.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_ocean.file = "@DATADIR@/toastd/templates/ptwc_ocean.html"

# ID of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_region.id = IDY0004

# Name of the template.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_region.name = "Region"

# Path to template file.
queues.production.processors.messages.toastd.bulletins.group.ptwc.template.ptwc_region.file = "@DATADIR@/toastd/templates/ptwc_region.html"
```

4.5.2 Initialize toast Database MariaDB

1. Login as sysop

```
root@host:~# mysql -u root -p
```

```
CREATE DATABASE seiscomp-tews CHARACTER SET utf8 COLLATE utf8_bin;
   grant usage on seiscomp-tews.* to sysop@localhost identified by 'sysop';
   grant all privileges on seiscomp-tews.* to sysop@localhost;
   grant usage on seiscomp-tews.* to sysop@'%' identified by 'sysop';
   grant all privileges on seiscomp-tews.* to sysop@'%';
   flush privileges;
   quit
```

2. Create the SeisComP database tables

```
sysop@host:~$ mysql -u sysop-p  toast  < /home/sysop/seiscomp-tews/share/db/mysql.sql
sysop@host:~$ mysql -U sysop-p  toast  < /home/sysop/seiscomp-tews/share/db/tsunami/mysql.sql
```

3. Restart module scmaster

```
sysop@host:~$ sc-tews  restart scmaster
```

4. Update configuration

```
sysop@host:~$ sc-tews  update-config
```