Installing SQream via Docker

SQream Technologies

Version 2019.2

Table of Contents

Table of Contents **3**
Installing SQream via Docker **5**
Before you start 5
Preinstallation procedures 5
Create and login as the SQream user 5
Configure the OS locale 5
Configure the time zone 5
Add the EPEL repository 6
Install required packages 6
Install recommended utilities (Optional) 6
Update to the latest release 7
Configure NTP 7
Configure the performance profile 7
Configure security limits 7
Tune kernel parameters 7
Set the fs.file-max value 8
Configuring the firewall 8
Install NVIDIA CUDA driver 10
Disable automatic bug reporting tools 11
Installation procedure 12
Download and run your Docker installer 12
Docker CE =18.03 12
CentOS: 12
Nvidia Docker2 12
CentOS: 12
Prepare the environment 12
Download the SQream package 12
Configure your environment 12
Flags 13

Comments 13
Example for best practice: 13
Setting data import folder example 14
Start the SQream Console 14
Console Commands 14
Metadata & Picker 14
SQreamd 14
List running services 14
Stop running services (Master and Workers) 15
Sqream Editor 15
SQream-Client 15
Moving from Docker to standard on-premises installation 16

Installing SQream via Docker

Before you start

Before you start the installation process, you should download your SQream package.

Contact SQream Technical Support to obtain your SQream package

Download the package to your server and untar it.

It will look like this:

```
sqream@ac922 sqream-console-package]$ ll
-rwxrwxr-x. 1 sqream sqream 10549 Mar 19 10:59 README.md
-rwxrwxr-x. 1 sqream sqream  1419 Mar 19 10:59 sqream-console
-rwxrwxr-x. 1 sqream sqream  8567 Mar 19 10:59 sqream-install
```

Preinstallation procedures

There are a number of preliminary steps that you must perform before you can actually install SQream.

- Creating the SQream user
- Configuring the OS
- Installing NVIDIA driver

Create and login as the SQream user

Run the following commands to create the sqream user and assign a password:

```
useradd -m -U sqream
passwd sqream
usermod -aG wheel sqream  ( can be removed from wheel after
installation is complete )
```

Now logout and login again as the sqream user.

Configure the OS locale

Set the system locale to *en_US.utf8:*

```
# sudo localectl set-locale LANG=en_US.UTF-8
```

Configure the time zone

Run the following command to display all the available time zones:

```
# timedatectl list-timezones
```

This will give you a list of the time zones available for your server. When you find the region/time zone setting that is correct for your server, run the following command to set it:

```
# sudo timedatectl set-timezone region/timezone
```

Example

To set the time zone to United States Eastern time, type:

```
# sudo timedatectl set-timezone America/New_York
```

Add the EPEL repository

```
sudo rpm -Uvh http://dl.fedoraproject.org/pub/epel/epel-release-latest-
7.noarch.rpm
```

Install required packages

```
sudo yum install ntp  pciutils python36.x86_64 monit kernel-devel-
$(uname -r) kernel-headers-$(uname -r)  gcc  tuned firewalld
```

Install recommended utilities (Optional)

The following packages contain tools that are recommended but not required for using SQream.

- **net tools:** tools for debugging
- **psmisc package:** utilities for managing processes

net-tools contain the following tools for debugging the network:

- ifconfig
- netstat

The psmisc package contains the following utilities for managing processes on your system:

- pstree: displays a tree structure of all of the running processes on your system
- killall: sends a specified signal (SIGTERM if nothing is specified) to processes identified by name
- fuser: identifies the PIDs of processes that are using specified files or filesystems

Run the following command to install the recommended tools

```
sudo yum install bash-completion.noarch vim-enhanced.x86_64 vim-
common.x86_64 net-tools iotop htop psmisc screen xfsprogs wget yum-utils
deltarpm dos2unix
```

Update to the latest release

```
sudo yum update
```

Configure NTP

NTP synchronization allows your computer to stay in sync with other servers, optimizing predictability in operations that rely on having the correct time.

If you have local NTP servers, add them to the NTP configuration.

```
sudo systemctl enable ntpd
sudo systemctl start ntpd
sudo ntpq -p
```

Configure the performance profile

```
sudo tuned-adm profile throughput-performance
sudo systemctl set-default multi-user.target
```

Configure security limits

On Linux systems, persistent limits can be set for a particular user by editing the */etc/security/limits.conf* file. To set the maximum number of open files for the sqream user to 500,000, run the following command to edit this *limits.conf* file.

```
# echo -e "sqream soft nproc 500000\nsqream hard nproc 500000\nsqream
soft nofile 500000\nsqream hard nofile 500000
```

Tune kernel parameters

Several kernel parameters can be tuned for better performance.

```
echo -e "vm.dirty_background_ratio = 5 \n vm.dirty_ratio = 10 \n
vm.swappiness = 10 \n vm.zone_reclaim_mode = 7 \n vm.vfs_cache_pressure
= 200 \n"  >> /etc/sysctl.conf
```

Check the fs.file max value.

```
sysctl -n fs.file-max
```

Set the fs.file-max value

If it is less than 2097152, run the following command.

```
echo "fs.file-max=2097152" >> /etc/sysctl.conf
```

Kernel Parameters

Parameter	Explanation
vm.vfs_cache_pressure	Controls the tendency of the kernel to reclaim the memory used for VFS caches versus pagecache and swap. Increasing this value increases the rate at which VFS caches are reclaimed.
vm.dirty_background_ratio	Percentage of total system memory. The number of pages at which the pdflush background writeback daemon will start writing out dirty data. However, for a fast RAID-based disk system this may cause large flushes of dirty memory pages. Larger values will result in less frequent flushes.
vm.swappiness	Defines how aggressively memory pages are swapped to disk. If you do not want swapping, then lower this value. However, if your system process sleeps for a long time you may benefit from an aggressive swapping behavior by increasing this value. You can change swappiness behavior by increasing or decreasing the value.
vm.dirty_ratio	Percentage of total system memory. The number of pages at which a process that is generating disk writes will itself start writing out dirty data. This is the ratio at which dirty pages created by application disk writes will be flushed out to disk. A value of 40 means that data will be written into system memory until the file system cache has a size of 40% of the server RAM. So if the server has 12GB of RAM, data will be written into system memory until the file system cache has a size of 4.8G.
vm.zone_reclaim_mode	Defines how aggressively memory will be reclaimed when a zone runs out of memory. Allowing zone reclaim to write out pages stops processes that are writing large amounts of data from dirtying pages on other nodes.
fs.file-max	file-max is the maximum number of File Descriptors (FD) enforced on a kernel level that cannot be exceeded by all combined processes without increasing. .

Configuring the firewall

If SQream DB runs on an internet-accessible server, we recommend that the public SQream database ports not be exposed to the internet. As with any application, the best strategy is to lock down everything that you do not have a good reason to keep open.

CentOS ships with a firewall called **firewalld**. The **firewalld** service has the ability to make modifications without dropping current connections, so you can turn it on before creating exceptions.

Run this command to start the firewalld service.

```
systemctl start firewalld
```

Now add a firewall "exception" to each SQream database port.

- If the server is to run the Metadata Server and Load Balancer, open ports 3105 and 3108.
- If the server is to run SQream database daemons, open the relevant ports for these daemons.

For example, for a server hosting all services and SQream database ports 5000-5003 run the following commands:

```
firewall-cmd --zone=public --permanent --add-port=2812/tcp
firewall-cmd --zone=public --permanent --add-port=3000/tcp
firewall-cmd --zone=public --permanent --add-port=3001/tcp
firewall-cmd --zone=public --permanent --add-port=3105/tcp
firewall-cmd --zone=public --permanent --add-port=3108/tcp
firewall-cmd --zone=public --permanent --add-port=5000-5003/tcp
firewall-cmd --zone=public --permanent --add-port=5100-5103/tcp
```

When you are finished, run the following command to see the list of the exceptions that will be implemented:

```
firewall-cmd --permanent --list-all
```

When you are ready to implement the changes, run this command to reload the firewall:

```
firewall-cmd --reload
```

If, after testing, everything works as expected, run this command to make sure the firewall will be started at boot:

```
systemctl enable firewalld
```

TIP:

You will have to explicitly open the firewall (with services or ports) for any additional services that you may configure in the future

Install NVIDIA CUDA driver

1. Reboot the servers.
2. Verify that your GPU is CUDA-capable.

```
lspci | grep -i nvidia
```

The expected result is something like:

```
06:00.0 3D controller: NVIDIA Corporation device [Tesla V100] (rev
a1)
07:00.0 3D controller: NVIDIA Corporation device [Tesla V100] (rev
a1)
```

If nothing appears, your GPU might not be installed correctly.

3. Make sure the upstream open-source nvidia driver (module) is not running.
   a. Check to see whether the upstream open-source nvidia driver (nouveau) is running:

```
lsmod | grep nouveau
```

   b. If it is running, disable it:

```
echo "blacklist nouveau" > /etc/modprobe.d/blacklist-
nouveau.conf
echo "options nouveau modeset=0"  >>
/etc/modprobe.d/blacklist-nouveau.conf
dracut --force
modprobe --showconfig | grep nouveau
```

   c. Reboot the server and verify that now the nouveau module is not loaded:

```
lsmod | grep nouveau
```

4. Install the CUDA repository.

```
sudo rpm -Uvh
https://developer.download.nvidia.com/compute/cuda/repos/rhel7/x86_
64/cuda-repo-rhel7-10.0.130-1.x86_64.rpm
```

OR

In advance,-prepare the cuda 10.0 offline repo - from a server that is connected to the cuda repo:

```
reposync -g -l -m --repoid=cuda --download_path=/var/cuda-repo-
10.0-local
```

Copy the repository to the installation server then run:

```
createrepo -g comps.xml  /var/cuda-repo-10.0-local
```

Add a repo configuration file in */etc/yum.repos.d/:*

```
vim /etc/yum.repos.d/cuda-10.0-local.repo
[cuda-10.0-local]
name=cuda-10.0-local
baseurl=file:///var/cuda-repo-10.0-local
enabled=1
gpgcheck=1
gpgkey=file:///var/cuda-repo-10.0-local/7fa2af80.pub
```

5. Install the actual CUDA:

```
sudo bash
yum install cuda-10-0.x86_64
```

6. Add some performance tuning to *rc.local*

```
vim /etc/rc.local
```

Add these lines to the file:

[V100]

```
nvidia-persistenced
```

[NOT V100]

```
nvidia-persistenced
nvidia-smi -pm 1
nvidia-smi -acp 0
nvidia-smi --auto-boost-permission=0
nvidia-smi --auto-boost-default=0
```

7. Reboot the server and run:

```
nvidia-smi
```

Disable automatic bug reporting tools

```
for i in abrt-ccpp.service abrtd.service abrt-oops.service abrt-
```

```
pstoreoops.service abrt-vmcore.service abrt-xorg.service ; do sudo
systemctl disable $i; sudo systemctl stop $i; done
```

You are now ready to install SQream.

Installation procedure

Download and run your Docker installer

Docker CE =18.03

CentOS:

https://docs.docker.com/install/linux/docker-ce/centos/

Nvidia Docker2

CentOS:

https://github.com/NVIDIA/nvidia-docker

CentOS 7 (docker-ce), RHEL 7.4/7.5 (docker-ce), Amazon Linux 1/2

Prepare the environment

SQream Docker runs on the 'docker' user.

In order for sqream to be able to read and write its resources, it's important that the docker user has r/w access to the following directories:

- Log (default: */var/log/sqream/*)
- Configuration (default: */etc/sqream/*)
- Cluster (where sqream writes the DB system. Example */mnt/sqreamdb*)
- Ingest (where the data that should be loaded is located. Example */mnt/data_source/*)

If running SQream from a user different then the docker user, the local user should be added to docker group:

```
sudo usermod -aG docker $USER
```

Logout or restart your ssh session.

Download the SQream package

Make sure your SQream package is downloaded and untarred. (See Before you start.)

Configure your environment

Run the commands below from the place where you put the package directory, for example */home/sqream/sqream-console-package/* under **sudo**:

```
./sqream-install -i -f -c -v -l -d
```

Flags

Flag	Explanation
l	Loads all the software from hidden folder ".docker" *(see Comment #3 below)*.
f	Overwrite if folder already exists, **Using f will overwrite all files in mounted directories**
c	Path *(see Comment #1 below)* where to write/read SQream configuration files from. (default: */etc/sqream/*).
v	Location of sqream cluster. If cluster doesn't exist it would be created, mounted if exists *(see Comment #3 below)*.
l	Location of SQream system startup logs (which contain startup logs and docker logs, Default: */var/log/sqream/*).
d	The folder where the customer currently has data to import/copy to sqream.
r	Reset system configuration, run without any other variables.
h	Help, shows available flags

Comments

1. If you are installing docker on a server that already works with SQream do not use the default path. (See Flag – **c**)
2. If you are not running the sqream-install for the first time, please verify that all SQream docker containers are stopped.
3. Mandatory flags for first run: (see flags –**i** and –**v**)

   ```
   sudo ./sqream-install -i -v /sqreamcluster
   ```

4. The "-**d**" flag is not required when installing sqream, however if the flag is not used, SQream will not be able to read from any folder except the SQream Cluster.

   If you did not use the "-**d**" flag in the original installation, you can run **sqream-install** with the -**d** flag at any time. Note that you will need to stop and start the **sqream worker** for the change to take effect.

   Example for best practice:

   ```
   sudo ./sqream-install -i -c /etc/sqream -v /home/sqream/sqreamdb -l
   /var/log/sqream -d /home/sqream/data_ingest
   ```

Setting data import folder example

```
sudo ./sqream-install -d /home/sqream/data_in
```

Start the SQream Console

```
./sqream-console
```

Console Commands

Metadata & Picker

Start metadata and picker

`sqream master --start --single-host`

```
sqream-console>sqream master --start --single-host
sqream_single_host_master is up and listening on ports: 3105, 3108
```

SQreamd

Start and stop SQream

`sqream worker --start X` (X is the number of GPUs you want to bind).

```
sqream-console>sqream worker --start 2
started sqream_single_host_worker_0 on port 5000, allocated gpu: 0
started sqream_single_host_worker_1 on port 5001, allocated gpu: 1
```

`sqream worker --stop <full worker name>` (run sqream master --list to get the worker name).

```
sqream-console>sqream worker --start 2
started sqream_single_host_worker_0 on port 5000, allocated gpu: 0
started sqream_single_host_worker_1 on port 5001, allocated gpu: 1
```

`sqream worker --stop <full worker name>` (run sqream master --list to get the worker name).

```
sqream-console>sqream worker --stop sqream_single_host_worker_0
stopped sqream_single_host_worker_0
```

List running services

View all running SQream services.

`sqream master --list`

```
sqream-console>sqream master --list
container name: sqream_single_host_worker_0, container id: c919e8fb78c8
container name: sqream_single_host_master, container id: ea7eef80e038
```

Stop running services (Master and Workers)

Stop all running SQream services

```
sqream master --stop
```

```
sqream-console>sqream master --stop
stopped container sqream_single_host_worker_0, id: 892a8f1a58c5
stopped container sqream_single_host_master, id: 55cb7e38eb22
```

Sqream Editor

Start SQream SQL Statement Editor

```
sqream editor --start
```

```
sqream-console>sqream editor --start
access sqream statement editor on your chrome web browser
http://192.168.0.220:3000
Stop SQream editor
```

```
sqream editor --stop
```

```
sqream-console>sqream editor --stop
stopped sqream_editor
```

SQream-Client

To use the embedded SQream Client on the master node:

- Default port 3108, use -p for other master port.
- Default database is master, use -d for different database) :

```
sqream client --master -u sqream -w sqream
```

To use the embedded SQream Client on worker node(s):

- Default port 5000, use -p for other worker port.
- Default database is master, use -d for a different database)

```
sqream client --worker -u sqream -w sqream
```

Moving from Docker to standard on-premises installation

Since Docker creates all files and directories on host as root, change sqream storage folder ownership to your working directory user.

Congratulations, you have successfully installed SQream DB and it is ready for use. We recommend the following SQream quick guides:

- Quick guide to launching a SQream DB cluster
- Quick guide to loading your data into a SQream database
- Quick guide to using the SQream Dynamic Workload Manager (DWLM)
- Quick guide to managing your SQream cluster