Advanced MySQL 15.03. - 19.03.2021

Agenda

1. Installation
    - Installation with tar-bundle Centos 8/Redhat 8
    - Installation from centos8/centos7
    - Login ohne passwort - mysql-client
    - Login with --login-path
    - Using HashiCorp-Vault for Authentication
2. Configuration
    - Port change on Centos/Redhat
3. Binlog - Management
    - Binlog
4. Backup and Restore
    - Best Practice mysqldump
    - Point-In-Time-Recovery
    - mysqlbackup - MySQL Enterprise Backup
    - xtrabackup 8 with Centos/Redhat
5. Variablen
    - Use-Cases Variables
6. MariaDB vs. MySQL
    - Comparing MySQL vs. MariaDB
7. Performance (InnoDB)
    - Important Configuration Settings-Innodb
8. Tools
    - Install percona toolkit - Centos/Redhat 8
    - Percona Monitoring and Slow Queries GUI
9. Indexes und slow query log
    - slow query log
    - Function Index
10. Replication
    - show slave hosts/show replicas
    - clone-plugin
    - Setup Replication with GTID
    - Installation Master-Slave with mysqlsh - mysql-shell
    - ReplicaSet - Troubleshoot
11. Group Replication/InnoDB Cluster
    - group replication

- innodb_cluster

12. Upgrade / Update

    - Upgrading from 5.7 to 8.0

13. Documentaton

    - MySQL Performance pdf
    - Incompabilities from MySQL 8 to MariaDB 10.5
    - multiple Bind-address starting from 8.0.13
    - pdf linux-security - selinux chapter
    - linux selinux type statements

Installation

Installation with tar-bundle Centos 8/Redhat 8

Steps, if there is not outside connection Linux Machine to MySQL repo or internet

```
## Step 1: Download bundle to your local machine
https://dev.mysql.com/downloads/mysql/

## Step 2: Download winscp
## just google

## Step 3: Transfer bundle to server with winscp


## Step 4: mv tar and untar folder
sudo su -
mv /home/kurs/mysql-8.0.23-1.el8.x86_64.rpm-bundle.tar /usr/src
cd /usr/src
mkdir mysql-install
cd mysql-install
tar xvf ../mysql-8.0.23-1.el8.x86_64.rpm-bundle.tar

## Step 5: change into folder and install necessary files
## Install no debuginfo - packages
yum install mysql-community-{server,client,common,libs,client-plugins}-8*
```

Start and retrieve password

```
systemctl start mysqld
systemctl enable mysqld
systemctl status mysqld
grep 'temporary password' /var/log/mysqld.log
```

Change temporary password

```
mysql -uroot -p
## important must fit password criteria
## using P@ssw0rd for training purpose
alter user root@localhost identified by 'P@ssw0rd';
```

Ref:

https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/linux-installation-rpm.html

Installation from centos8/centos7

```
yum install mysql-server mysql
```

Login ohne passwort - mysql-client

Simple version with cleartext pass

```
## cat /root/.my.cnf
[client]
password=P@ssw0rd
## now you can call as root

mysql
```

Login with --login-path

Walkthrough - Default (client)

```
## Step 1: Eventually Set/Change password
mysql>alter user root@localhost identified by 'MYSECRETPASS'
mysql>quit
## set pass in loginpath
## in this case if not set loginpath is 'client'

## Step  2: set password for client use
## Set the same pass as set above
[root@localhost ~]# mysql_config_editor set --user=root --password

## Step 3: use it
## Feel happy using / pass will be taken from:
[root@localhost ~]# ls -la /root/.mylogin.cnf
[root@localhost ~]# mysql
```

Walkthrough - loginpath=admin (client)

```
## Step 1: Eventually Set/Change password
mysql>alter user admin@localhost identified by 'MYSECRETPASS'
mysql>quit
## set pass in loginpath
## in this case if not set loginpath is 'client'

## Step  2: set password for client use
## Set the same pass as set above
[root@localhost ~]# mysql_config_editor set --loginpath=admin --user=root --password

## Step 3: use it
## Feel happy using / pass will be taken from:
[root@localhost ~]# ls -la /root/.mylogin.cnf
[root@localhost ~]# mysql --loginpath=admin
```

Side - Notes (file is not encrypted but obfuscated)

- https://jira.mariadb.org/browse/MDEV-20665
- https://mariadb.com/kb/en/mysql_config_editor-compatibility/

How to see set password in .mylogin.cnf (unsafe)

```
## https://www.percona.com/blog/2016/09/07/get-passwords-plain-text-mylogin-cnf/
my_print_defaults -s client
```

Ref:

- https://dev.mysql.com/doc/refman/8.0/en/mysql-config-editor.html

Using HashiCorp-Vault for Authentication

- https://www.vaultproject.io/docs/secrets/databases

Configuration

Port change on Centos/Redhat

```
## Walkthrough
/etc/my.cnf
port=13306

## Change port in selinux
## this will be persistent across reboots
semanage port -a -t mysqld_port_t -p tcp 13306

## Errors can be found in
## /var/log/audit/audit.log

systemctl restart mysqld
```

Binlog - Management

Binlog

Key Facts

```
show variables like '%log_bin%';
## set in configuration
log_bin
## changed during session for not logging actions in session to binlog
sql_log_bin
```

show master status

```
## at which position is the master currently
show master status
```

when using slave potential master ....

```
## on by default on mysql 8
## off by default on mysql 5.7
log_slaves_updates = on

## this must be on, if you want to use slave as master later
## alle update from master are logged to binary of slave if log-bin=on
```

Disable binary logging

```
## /etc/my.cnf
[mysqld]
skip-log-bin
## or
## disable-log-bin

### Restart
## systemctl restart mysqld
## now master is empty
mysql>show master status
```

binlog_format

```
## STATEMENT
## The direct statement will be used
## e.g. INSERT INTO actors (first_name, last_name) values ('hans','mustermann')

## MIXED
## Server decides if to use STATEMENT or ROW for each sql - statement

## ROW
```

Workaround for tail -f ;o) for binlog

```
mysqlbinlog -vvv -R -t --stop-never binlog.00000
```

mysqlbinlog with date

```
mysqlbinlog --stop-datetime="2021-03-15 15:23" binlog.000012
```

mysqlbinlog over multiple binlogs

```
mysqlbinlog -vvv --stop-position=465 binlog.000010 binlog.000011 binlog.000012 >
/usr/recovery.sql
```

Ref:

https://dev.mysql.com/doc/refman/8.0/en/replication-options-binary-log.html

Backup and Restore

Best Practice mysqldump

best practice mysqldump

```
## important to dump --event and --routines starting from
## mysql 8, because they are not part of mysql-tables anymore

## flush-logs - create new binlog after dump -> good for pit (Point-In-Time -
Recovery)
## master-data = 2 (2 means comment)
##
mysqldump --all-databases --flush-logs --master-data=2 --routines --events >
/usr/src/all-databases.sql
```

Also delete old logs after mysqldump (not for master-slave)

```
mysqldump --delete-master-logs --all-databases --flush-logs --master-data=2 --routines
--events > /usr/src/all-databases.sql
```

use transaction -> for dumping

```
mysqldump --single-transactions --delete-master-logs --all-databases --flush-logs --
master-data=2 --routines --events > /usr/src/all-databases.sql
```

Point-In-Time-Recovery

mysqlbackup - MySQL Enterprise Backup

Walkthrough

```
## Download von mysqlbackup
### Step 1
## File auf ssh-server runterlasen (MySQL 2) und installieren
sudo su -
cd /usr/src
unzip V1006234-01.zip
rpm -i mysql-commercial-backup-8.0.23-1.1.el8.x86_64.rpm

### Step 2
## Backup durchführen
mysqlbackup --user=root --password --host=127.0.0.1 --backup-image=/backups/my.mbi \
  --backup-dir=/backups/backup-tmp  backup-to-image
echo $? # 0 is success

### Step 3
mysqlbackup --backup-image=/backups/my.mbi validate
echo $?

### Step 4
## systemctl stop mysqld
cd /var/lib/
mv mysql mysql.bkup
### backup-tmp-back needs to be empty
mysqlbackup --backup-image=/backups/my.mbi --backup-dir=/backups/backup-tmp-back copy-
back-and-apply-log
chown -R mysql:mysql /var/lib/mysql
systemctl start mysqld
```

xtrabackup 8 with Centos/Redhat

Walkthrough

```
xtrabackup --backup --target-dir=/backups/20210316
## Apply crash-recovery-log -> aka ib_logfile_1
xtrabackup --prepare --target-dir=/backups/20210316
## Restore
systemctl stop mysqld
cd /var/lib
mv mysql mysql.bkup
xtrabackup --copy-back --target-dir=/backups/20210316
chown -R mysql:mysql mysql
## special selinux
restorecon -r /var/lib/mysql
systemctl start mysqld
```

move-back with alternative datadir

```
xtrabackup --copy-back --datadir=/var/lib/coolio --target-dir=/backups/20210316
```

Variablen

Use-Cases Variables

Select into

```
select first_name,last_name from actor where actor_id = 1;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| PENELOPE   | GUINESS   |
+------------+-----------+
1 row in set (0.00 sec)

mysql> select first_name,last_name into @vorname,@nachname from actor where actor_id =
1;
Query OK, 1 row affected (0.00 sec)

mysql> select @vorname;
+----------+
| @vorname |
+----------+
| PENELOPE |
+----------+
1 row in set (0.00 sec)

mysql>
```

Variables can only be used for field-values

```
### This does not work

mysql> set @tabellenname='actor'
mysql> insert into @tabellenname (first_name,last_name) values ('a','b');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near
'@tabellenname (first_name,last_name) values ('a','b')' at line 1
mysql> insert into concat(@tabellenname,'') (first_name,last_name) values ('a','b');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near
'@tabellenname,'') (first_name,last_name) values ('a','b')' at line 1
mysql> select concat(@tabellenname,'');
+--------------------------+
| concat(@tabellenname,'') |
+--------------------------+
| actor                    |
+--------------------------+
1 row in set (0.00 sec)
```

```
mysql> set @feld='last_name';
Query OK, 0 rows affected (0.00 sec)

mysql> insert into tabellenname (@feld, first_name) values ('a','b');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '@feld,
first_name) values ('a','b')' at line 1
mysql> insert into tabellenname (last_name, first_name) values (@feld,'b');
ERROR 1146 (42S02): Table 'sakila.tabellenname' doesn't exist
```

```
## THIS DOES WORK
mysql> insert into actor (last_name, first_name) values (@feld,'b');
Query OK, 1 row affected (0.00 sec)

mysql> insert into @tabellenname (last_name, first_name) values (@feld,'b');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near
'@tabellenname (last_name, first_name) values (@feld,'b')' at line 1
```

MariaDB vs. MySQL

Comparing MySQL vs. MariaDB

Cluster

```
MariaDB (im Bauch) - im Repo:
=============================
2007:
Galera Cluster (virtuell synchrone Cluster)
--> Auto-Provisioning -> Node konfigurieren -> dann holt sie sich die Daten selbst

MySQL - im Repo:
================
viel, viel später:
Group Replication (=Cluster)

aber: es gibt auch eine binary von codership - mit galera cluster
```

Features

- Welche Features brauche ich .

Spezielle Features von MariaDB

- Flashback (auf älteren Datenstand gehen ohne komplettes neues Backup + PIT - Recovery.
- Encryption

Speziele von MySQL

- Persistent settings of variables
- Spezielle Tools (mysql_repl)

Unterschiede sind

- Implementierungen ändern sich (z.B. mariadb -> user -> views)

MariaDB

- MaxScale + MariaDB Galera Cluster

Performance (InnoDB)

Important Configuration Settings-Innodb

innodb_buffer_pool_size

```
## how big is it ?
show variables like 'innodb_buffer%';

## 70-80% of memory of system
## evaluate with
show engine innodb status
pager grep -i 'free buffers'
show engine innodb status
pager
```

innodb_log_buffer_size

```
## how big is it ?
show variables like 'innodb_log_buffer%'

### should be able to hold one commit
```

innodb_log_file_size

```
## Should hold work load of 120 minutes
```

- https://www.percona.com/blog/2008/11/21/how-to-calculate-a-good-innodb-log-file-size/

innodb_flush_trx_commit

```
## Defaults to 1
## flush from log_buffer to log
## after every commit
## ACID compliant

## Only set to 0 or 2 if you are willing to loose 1 second of data
## in case of Stand-Alone and Master-Slave

## On group-replication / galera cluster
## 0 <- is safe because of virtuell synchronisation
```

innodb_flush_method

```
##### Linux
O_DIRECT
## use only on linux,
## can increase performance

### Do not use  or be critical about it
O_DIRECT_NOI_FSYNC
```

```
### Set persistent and restarte server
## only persist do not change during runtime
mysql> set persist_only innodb_flush_method = 'O_DIRECT';
## only works when systemd is the start/stop - system
## or windows ervice
mysql> restart
## now it is changed
mysql> show variables like 'innodb_flush_method'

##### Windows

## no need to change, because O_DIRECT does not work
```

innodb_flush_neighbors

```
## for ssd - disks keep 0 which is default

## for hd - set to 1
## means dirty neighbor pages are flushed to disk a well from
## innodb_buffer_pool on flushing
set persist innodb_flush_neighbors=1;
```

skip-name-resolve

```
## do not do name resolving / no local or dns lookup
## from now on user@hostname entries do not work
mysql> set persist_only skip_name_resolve = ON;
mysql> restart
```

Tools

Install percona toolkit - Centos/Redhat 8

Walkthrough

```
yum install https://repo.percona.com/yum/percona-release-latest.noarch.rpm
yum search percona
yum install percona-toolkit
## now tools are available starting with pt-
pt- # TAB - TAB
```

Percona Monitoring and Slow Queries GUI

Demo

- https://pmmdemo.percona.com/

Documentation

- https://www.percona.com/doc/percona-monitoring-and-management/2.x/index.html

Indexes und slow query log

slow query log

Activation

```
mysql> set global slow_query_log = on
-- in seconds
mysql> set global long_query_time = 0.01
-- log sql - staments that have no index
mysql> set global log_queries_not_using_indexes = ON
```

Function Index

Replication

show slave hosts/show replicas

```
## from 5.7. -> 8.0.21
show slave hosts
## from 8.0.22
show replicas
```

clone-plugin

Walkthrough

```
## On Joiner -> Server that shall get the clone
## Plugin needs to loaded
mysql> show plugins
## if not present install
mysql> install plugin clone soname 'mysql_clone.so'

## On Donor - server that provides clone
## needs a user with 'BACKUP_ADMIN'
## Joiner: needs a 'CLONE_ADMIN'
## on Donor -> master
create user cloneuser@'192.168.56.105' identified by 'your_Secret_pass';
grant backup_admin,clone_admin on *.* to cloneuser@'192.168.56.105';

## On Joiner -  test connection
mysql -ucloneuser -p -h 192.168.56.105
mysql> show grants
```

Prerequisites for cloning

```
## on both systems these values must be identical
## collation / character - the server stuff
mysql> show variables like 'innodb_page_size';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| innodb_page_size | 16384 |
+------------------+-------+
1 row in set (0.00 sec)

mysql> show variables like 'innodb_data_file_path';
+-----------------------+------------------------+
| Variable_name         | Value                  |
+-----------------------+------------------------+
| innodb_data_file_path | ibdata1:12M:autoextend |
+-----------------------+------------------------+
1 row in set (0.00 sec)

mysql> show variables like 'max_allowed_packet';
+--------------------+----------+
| Variable_name      | Value    |
+--------------------+----------+
| max_allowed_packet | 67108864 |
+--------------------+----------+
1 row in set (0.01 sec)

mysql> show global variables like '%character%';
+--------------------------+----------------------------+
| Variable_name            | Value                      |
```

```
+--------------------------+----------------------------+
| character_set_client     | utf8mb4                    |
| character_set_connection | utf8mb4                    |
| character_set_database   | utf8mb4                    |
| character_set_filesystem | binary                     |
| character_set_results    | utf8mb4                    |
| character_set_server     | utf8mb4                    |
| character_set_system     | utf8                       |
| character_sets_dir       | /usr/share/mysql/charsets/ |
+--------------------------+----------------------------+
8 rows in set (0.01 sec)

mysql> show global variables like '%collation%';
+--------------------------------+--------------------+
| Variable_name                  | Value              |
+--------------------------------+--------------------+
| collation_connection           | utf8mb4_0900_ai_ci |
| collation_database             | utf8mb4_0900_ai_ci |
| collation_server               | utf8mb4_0900_ai_ci |
| default_collation_for_utf8mb4 | utf8mb4_0900_ai_ci |
+--------------------------------+--------------------+
4 rows in set (0.01 sec)

mysql> select tablespace_name, file_name from information_schema.files where file_type
like 'UNDO LOG';
+-----------------+------------+
| TABLESPACE_NAME | FILE_NAME  |
+-----------------+------------+
| innodb_undo_001 | ./undo_001 |
| innodb_undo_002 | ./undo_002 |
+-----------------+------------+
2 rows in set (0.00 sec)
```

Increase Verbosity

```
## on master - really important
## because you can see the progress and return-code: success -> 0
mysql> set global log_error_verbosity=3

## eventually on joiner - not tested output verbosity yet
mysql> set global log_error_verbosity.de
```

Overview clone variables

```
 show variables like '%clone%'
    -> ;
+------------------------------+---------+
| Variable_name                | Value   |
+------------------------------+---------+
| clone_autotune_concurrency   | ON      |
```

```
| clone_buffer_size            | 4194304 |
| clone_ddl_timeout            | 300     |
| clone_enable_compression     | OFF     |
| clone_max_concurrency        | 16      |
| clone_max_data_bandwidth     | 0       |
| clone_max_network_bandwidth | 0       |
| clone_ssl_ca                 |         |
| clone_ssl_cert               |         |
| clone_ssl_key                |         |
| clone_valid_donor_list       |         |
+------------------------------+---------+
11 rows in set (0.01 sec)
```

Set compression on master (but takes cpu-load)

```
## before cloning or in general
set global clone_enable_compression = ON
```

Setup donor list and start clone

```
## on joiner
mysql> set global clone_valid_donor_list = '192.168.56.103:3306';
mysql> show variables like 'clone_valid_donor_list';

mysql> CLONE INSTANCE FROM 'cloneuser'@'192.168.56.103':3306 identified by 'P@ssw0rd';

mysql> show variables like 'clone_valid_donor%';
+------------------------+-------+
| Variable_name          | Value |
+------------------------+-------+
| clone_valid_donor_list |       |
+------------------------+-------+
```

Setup Replication with GTID

On master setup settings

```
mysql>set persist server_id=1;
mysql>set persist_only gtid_mode=ON
mysql>set persist_only enforce_gtid_consistency=true;
mysql>restart;
```

Setup replication user

```
create user 'repl'@'%' identified by 'mysecrectpass' require ssl;
grant replication slave on *.* to 'repl'@'%'
```

On Slave use clone plugin

- See entry in this documentation

On Slave

```
mysql>set persist server_id=2;
mysql>set persist_only gtid_mode=ON
mysql>set persist_only enforce_gtid_consistency=true;
mysql>restart

set persist server_id=2;
set persist_only gtid_mode=ON;
set persist_only enforce_gtid_consistency=true;
restart;
```

Now setup master-connection-config

```
change master to master_host='192.168.56.103',
master_user='repl',master_password='mysecretpass',
master_auto_position=1, master_ssl = 1;
start slave;
show slave status \G
```

Installation Master-Slave with mysqlsh - mysql-shell

Remarks

```
## Warning: Do not try to use mysql-shell from epel-repo -> Centos/Redaht
## javascript is not supported
```

Installation on both servers

```
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
yum install mysql-shell
```

Configuration Basis Configuration on both servers

```
mysqlsh --uri root@localhost
dba.configureReplicaSetInstance()
## use 2
## setup user: repl
## and agree to do the changes -> Y
## Restart : Y
```

Now login in as replication user on master

```
mysql --uri repl@master
rs=dba.createReplicaSet('MasterSlaveGroup')
rs.status()
```

Probably if not reachable (test with telnet) setup firewalld

```
## test if ports 3306 and 33061 are reachable from outside
## if not
firewall-cmd --add-service=mysql --permanent
firewall-cmd --add-port=33060/tcp --permanent
firewall-cmd --reload
```

Setup mysqlrouter

```
## on mysqlrouter server / same as app-server
## set same /etc/hosts file

192.168.56.103 master.training.local master
192.168.56.105 slave.training.local slave
192.168.56.106 router.training.local router

yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
yum install mysql-router mysql

mysqlrouter --bootstrap repl@slave --conf-use-sockets --user mysqlrouter
systemctl start mysqlrouter
```

Setup on master

```
## mysql
mysql>create user training@'192.168.56.%' identified by 'P@ssw0rd';
mysql>grant all on *.* to training@'192.168.56.%';
```

ReplicaSet - Troubleshoot

Node crashes and data was changed

```
## node cannot join automatically
## you need to set the recoveryMethod
rs=dba.getReplicatSet()
rs.status()
rs.rejoinInstance('repl@slave',{recoveryMethod: 'clone'})
```

Group Replication/InnoDB Cluster

group replication

Setup Server 1

```
Step 0.5:
yum install wget
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
dnf remove @mysql
dnf module reset mysql && dnf module disable mysql
cd /etc/yum.repos.d
rm -fR percona*
yum install mysql-server
```

```
## Step 1: set pw

## Step 2: Get uuid for whole group
mysql> select uuid()

## Step 3: Create /etc/my.cnf.d/z_cluster.cnf
## + include includedir /etc/my.cnf.d
## cat /etc/my.cnf
includedir /etc/my.cnf
```

```
## Step 4 setup firewall
firewall-cmd --zone=public --list-all
firewall-cmd --add-service=mysql --permanent
firewall-cmd --add-port=33060-33061/tcp --permanent
firewall-cmd --reload

## Step 4a: selinux adjustments ports
semanage -a -t mysqld_port_t -p tcp 33061
```

```
## Step 5: setting z_cluster.cnf
[mysqld]

server_id=1
bind-address=0.0.0.0
gtid_mode=ON
enforce_gtid_consistency=ON
## from 8.0.23 on, mysql understand checksum
## so no need to set to NONE
## binlog_checksum=NONE

plugin_load_add='group_replication.so'
group_replication_single_primary_mode=OFF
loose-group_replication_group_name="9d7a361f-8884-11eb-9199-525400278b50"
loose-group_replication_start_on_boot=OFF
loose-group_replication_local_address= "192.168.56.102:33061"
loose-group_replication_group_seeds="192.168.56.102:33061, 192.168.56.103:33061,
```

```
192.168.56.104:33061"
loose-group_replication_bootstrap_group=OFF
report_host=192.168.56.102

## restart
systemctl restart mysqld
```

```
## Step 5: Setup replication user
## within mysql> client
set sql_log_bin = 0;
create user repl@'192.168.56.%' identified by 'P@ssw0rd';
grant replication slave on *.* to 'repl'@'192.168.56.%';
set sql_log_bin=1;
change master to master_user='repl', master_password='P@ssw0rd' for channel
'group_replication_recovery';
```

```
## Step 6: Start group replication of server 1 (bootstrap)
## within mysql - client
set global group_replication_bootstrap_group=on;
start group_replication;
set global group_replication_bootstrap_group=off;
```

```
## Step 7: Show status
select * from performance_schema.replication_group_members \G
```

Setup Server 2

```
## Step 0.5:
yum install wget
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
dnf remove @mysql
dnf module reset mysql && dnf module disable mysql
cd /etc/yum.repos.d
rm -fR percona*
yum install mysql-server
```

```
## Start and set root pw
systemctl start mysqld
```

```
## Step 4 setup firewall
firewall-cmd --zone=public --list-all
firewall-cmd --add-service=mysql --permanent
firewall-cmd --add-port=33060-33061/tcp --permanent
firewall-cmd --reload

## Step 4a: selinux adjustments ports
semanage port -a -t mysqld_port_t -p tcp 33061
```

Setup Server 3

```
## Step 0.5:
yum install wget
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
dnf remove @mysql
dnf module reset mysql && dnf module disable mysql
cd /etc/yum.repos.d
rm -fR percona*
yum install mysql-server
```

```
## Start and set root pw
systemctl start mysqld
```

```
## Step 4 setup firewall
firewall-cmd --zone=public --list-all
firewall-cmd --add-service=mysql --permanent
firewall-cmd --add-port=33060-33061/tcp --permanent
firewall-cmd --reload

## Step 4a: selinux adjustments ports
semanage port -a -t mysqld_port_t -p tcp 33061
```

innodb_cluster

Pre-Requisites:

```
Install 4 Servers:
e.g.
192.168.56.101  gr1.training.local gr1
192.168.56.102  gr2.training.local gr2
192.168.56.103  gr3.training.local gr3
192.168.56.104  router.training.local router

- set hostname properly for each server:
hostnamectl set-hostname gr1.training.local

on every server add the above configuration to the /etc/hosts
```

Step 1: Setup Server 1

```
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
yum install mysql-server mysql mysql-shell

#### set firewall and selinux
semanage port -a -t mysqld_port_t -p tcp 33061
firewall-cmd --add-service=mysql
firewall-cmd --add-port 33060-33061/tcp --permanent
firewall-cmd --reload

#### start and enable service
systemctl start mysqld
systemctl enable mysqld

## set pw
cat /var/log/mysqld.log | grep temp
mysql -p
mysql> create user root@localhost identified by 'P@ssw0rd'

#### set firewall and selinux
semanage port -a -t mysqld_port_t -p tcp 33061
firewall-cmd --add-service=mysql
firewall-cmd --add-port 33060-33061/tcp --permanent
firewall-cmd --reload

### configure the instance
mysqlsh --uri=root@localhost
JS> dba.configureInstance()
```

Step 2: Setup Server 2

```
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
yum install mysql-server mysql mysql-shell
```

```
#### set firewall and selinux
semanage port -a -t mysqld_port_t -p tcp 33061
firewall-cmd --add-service=mysql
firewall-cmd --add-port 33060-33061/tcp --permanent
firewall-cmd --reload

#### start and enable service
systemctl start mysqld
systemctl enable mysqld

## set pw
cat /var/log/mysqld.log | grep temp
mysql -p
mysql> create user root@localhost identified by 'P@ssw0rd'

#### set firewall and selinux
semanage port -a -t mysqld_port_t -p tcp 33061
firewall-cmd --add-service=mysql
firewall-cmd --add-port 33060-33061/tcp --permanent
firewall-cmd --reload

### configure the instance
mysqlsh --uri=root@localhost
JS> dba.configureInstance()
```

Step 3: Setup Server 3

```
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
yum install mysql-server mysql mysql-shell

#### set firewall and selinux
semanage port -a -t mysqld_port_t -p tcp 33061
firewall-cmd --add-service=mysql
firewall-cmd --add-port 33060-33061/tcp --permanent
firewall-cmd --reload

#### start and enable service
systemctl start mysqld
systemctl enable mysqld

## set pw
cat /var/log/mysqld.log | grep temp
mysql -p
mysql> create user root@localhost identified by 'P@ssw0rd'

#### set firewall and selinux
semanage port -a -t mysqld_port_t -p tcp 33061
firewall-cmd --add-service=mysql
firewall-cmd --add-port 33060-33061/tcp --permanent
firewall-cmd --reload

### configure the instance
```

```
mysqlsh --uri=root@localhost
dba.configureInstance()

### --> add User -> [2]
### clusteradmin
### pw:  yourclusteradminpw
```

Step 4: Setup Cluster and addInstances (on gr1- server1)

```
mysqlsh --uri clusteradmin@gr1
JS> rs.dba.createCluster('devCluster')
JS> rs.status()
JS> rs.addInstance('clusteradmin@gr2')
JS> rs.addInstance('clusteradmin@gr3')
```

Step 5: Setup mysqlrouter on Server 4

```
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
yum install mysql mysqlrouter
mysqlrouter --bootstrap clusteradmin@gr2  --user mysqlrouter  --force
systemctl start mysqlrouter
systemctl enable mysqlrouter
```

Upgrade / Update

Upgrading from 5.7 to 8.0

Step 1: For lab - starting with installing MySQL 5.7.

```
## As mysql 5.7. is not present in Centos 8 repo-rpm -file
## Use the one for Centos 7
yum install https://dev.mysql.com/get/mysql80-community-release-el7-3.noarch.rpm

## disable mysql from centos repo
dnf remove @mysql
dnf module reset mysql && dnf module disable mysql
dnf config-manager --disable mysql80-community
dnf config-manager --enable mysql57-community
yum install mysql-community-server

##
systemctl start mysqld
## get temporary pass
cat /var/log/mysqld.log | grep temp
mysql -p
## Change pass in mysql
mysql>alter user root@localhost identified by 'mysecretpass'
mysql_config_editor set --user=root --password
```

Step 1b (Optiona): Install sakila

```
yum install wget
cd /usr/src
wget https://downloads.mysql.com/docs/sakila-db.tar.gz
tar xzvf sakila-db.tar.gz
cd sakila-db
mysql < sakila-schema.sql
mysql < sakila-data.sql
```

Step 2: Uninstall old repo and install new repo

```
yum remove mysql80-community-release
yum install https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
```

Step 3: Install MySQL 8 - Shell

```
yum install mysql-shell
```

Step 4: Run checks

```
mysqlsh
JS> shell.connect('root@localhost')
JS> util.checkForServerUpgrade({configPath: '/etc/my.cnf'})
```

Step 5: Create Backup of Data with mysqldump

```
mysqldump --all-dabases > /usr/src/dump57.sql
echo $? # Was dump succesful
```

Step 6:

```
## stop server
systemctl stop mysqld
## uninstall server
yum list installed | grep mysql
yum remove mysql-community-server

## Install new mysql 8 server
## works because we already have the new repo in place
yum install mysql-community-server

## now start the new server and see, if it will come up
## this will be an INPLACE update, which is recommended by MySQL
## if everything works well the system will come up again
systemctl start mysqld

## look into the logs and status if mysqld is ready
tail /var/log/mysqld.log
systemctl status mysqld
```

Help for check-command

```
MySQL  JS > \? checkForServerUpgrade
```

Documentaton

MySQL Performance pdf

- http://schulung.t3isp.de/documents/pdfs/mysql/mysql-performance.pdf

Incompabilities from MySQL 8 to MariaDB 10.5

- https://mariadb.com/kb/en/incompatibilities-and-feature-differences-between-mariadb-105-and-mysql-80/

multiple Bind-address starting from 8.0.13

- https://mysqlserverteam.com/the-bind-address-option-now-supports-multiple-addresses/

pdf linux-security - selinux chapter

- http://schulung.t3isp.de/documents/linux-security.pdf

linux selinux type statements

- https://selinuxproject.org/page/TypeStatements