Beyond apt install
Building Safe MySQL Test Environments

Sveta Smirnova
Percona

PERCONA

Connecting to EC2

PERCONA

Download Key

https://tinyurl.com/PL2026-sveta

PERCONA

Fix Permissions

```
chmod 400 PL2026.pem
```

Connect

```
ssh -i "PL2026.pem" ubuntu@YOUR_URL
```

Verify

```
$ ls
Percona-Server-8.4.8-8-Linux.x86_64.glibc2.28.tar.gz
anydbver dbdeployer-ProxySQL dbdeployer-beto
libssl1.1_1.1.1f-1ubuntu2_amd64.deb
mysql-9.7.0-linux-glibc2.28-x86_64.tar.xz
mysql-test-9.7.0-linux-glibc2.28-x86_64.tar.xz
test_db
```

Installing from Tarball

PERCONA

Scenarios

- Upstream MySQL
- Percona Server for MySQL
- Replication between Two Servers
- Fancy setup

DBdeployer

Actual Versions

github.com/Robertoh98/dbdeployer

github.com/ProxySQL/dbdeployer

Scenarios

- Upstream MySQL
- Percona Server for MySQL
- Replication between Two Servers
- Fancy setup

Anydbver

PERCONA

Download

github.com/zelmario/anydbver

PERCONA

Scenarios

- Upstream MySQL
- Percona Server for MySQL
- Replication between Two Servers
- Loop
  1. Exploring options
  2. Fancy setup

Docker

PERCONA

Versions

hub.docker.com/_/mysql

hub.docker.com/r/percona/percona-server

PERCONA

Run MySQL

```
docker run -it -p 33844:3306
-d --name mysql-9.7.0
-e MYSQL_ROOT_PASSWORD=My_password111!
mysql:9.7.0
```

Connect

```
mysql -h127.0.0.1 -P33844 -uroot
    -p -e 'SELECT VERSION()'
```


PERCONA

Run Percona

```
docker run -it -p 33845:3306
-d --name ps-8.4
-e MYSQL_ROOT_PASSWORD=My_password111!
percona/percona-server:8.4
```


PERCONA®

Connect

```
mysql -h127.0.0.1 -P33845 -uroot -p
```

Your Own Images

```
mkdir docker-test
cd docker-test
vim config-file.cnf
cp -R /path/to/mysql/data .
```


PERCONA

Custom Dockerfile

```
FROM mysql:9.7.0

COPY config-file.cnf /etc/mysql/conf.d/

COPY data /var/lib/mysql

EXPOSE 3306

CMD ["mysqld"]
```


PERCONA

Create the Image

```
docker build -t custom-mysql:9.7.0
.
```


PERCONA

Start container

```
docker run -it -p 33846:3306
  --name mysql-9.7.0-custom
  -d custom-mysql:9.7.0
```

Connect

```
mysql -h127.0.0.1 -P33846 -uroot -p
```

MTR (advanced)

SQL you are used to: mtr_test_1.test

```
#--source include/have_innodb.inc;

CREATE TABLE `t` (
  `id` int(11) NOT NULL,
  `f` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

insert into t values(12345, 'value1'), (54321, 'value2');
select * from t;
show create table t;

drop table if exists t;
```


PERCONA

Custom Options

```
$ cat suite/mtr_test/t/mtr_test_2-master.opt
--transaction-isolation=read-committed
```


PERCONA

Concurrent Execution

```
CREATE TABLE ‘t‘ (
  ‘id‘ int(11) NOT NULL,
  ‘f‘ varchar(100) DEFAULT NULL,
  PRIMARY KEY (‘id‘)
  ) ENGINE=InnoDB;
insert into t values(12345, ’value1’),
                    (54321, ’value2’);

begin;
select * from t;
```

```
--connect(addconroot,localhost,root,,)
--connection addconroot

begin;
update t set f=’foo’ where id=12345;
commit;

--connection default
select * from t;

drop table if exists t;
```


PERCONA

Error Handling

```
CREATE TABLE ‘t‘ (
  ‘id‘ int(11) NOT NULL,
  ‘f‘ varchar(100) DEFAULT NULL,
  PRIMARY KEY (‘id‘))
ENGINE=InnoDB;
insert into t values(12345, ’value1’),
                    (54321, ’value2’);
begin;
select * from t where id=12345
lock in share mode;
```

```
--connect(addconroot,localhost,root,,)
--connection addconroot

set innodb_lock_wait_timeout=3;
begin;
--error 1205
update t set f=’value3’ where id=12345;
rollback;

--connection default
rollback;
drop table if exists t;
```


PERCONA

Result Processing

```
...
--connect(addconroot,localhost,root,,)
--connection addconroot
begin;
select * from t where id=54321 for update;

--connection default
--send update t set f='value3' where id=54321

--connection addconroot
update t set f='value3' where id=12345;

--connection default
--error 1213
--reap
```


PERCONA

External Commands

```
...
--exec $MYSQL_DUMP test > $MYSQL_TEST_DIR/var/tmp/mtr_amer_5.sql
--exec ls $MYSQL_TEST_DIR/var/tmp/

drop table t;
show tables;

--exec $MYSQL test < $MYSQL_TEST_DIR/var/tmp/mtr_amer_5.sql
show tables;
select * from t;
...
```


PERCONA

Flow Control

```
CREATE TABLE `t` (
  `id` int(11) NOT NULL,
  `f` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

--disable_query_log
--let $c=1
while ($c<100)
{
--eval insert into t values($c, md5($c))
--inc $c
}
--enable_query_log
```


PERCONA

Replication out of the Box

```
$ cat suite/mtr_test/t/rpl_mtr_test_7.test

#--source include/master-slave.inc
--source include/rpl/init_source_replica.inc

# We are on source
CREATE TABLE `t` (
  `id` int(11) NOT NULL,
  `f` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;
insert into t values(12345, 'value1'), (54321, 'value2');
select * from t;
show create table t;
```


PERCONA

Replication out of the Box

```
--connection slave
--vertical_results
show replica status;

--connection master
--horizontal_results
update t set f='foo';
```


PERCONA

Replication out of the Box

```
#We are on source
--sync_slave_with_master
#We are on replica
select * from t;

--connection source
drop table if exists t;

--sync_slave_with_master
stop replica;
```


PERCONA

Replication Options

- On source

```
$ cat suite/mtr_test/t/rpl_mtr_test_7-master.opt
--gtid_mode=ON  --log-slave-updates --enforce-gtid-consistency --binlog-
format=row
```


PERCONA

Replication Options

- On source

```
$ cat suite/mtr_test/t/rpl_mtr_test_7-master.opt
--gtid_mode=ON  --log-slave-updates --enforce-gtid-consistency --binlog-
format=row
```

- On replica

```
$ cat suite/mtr_test/t/rpl_mtr_test_7-slave.opt
--gtid_mode=ON  --log-slave-updates --enforce-gtid-consistency --binlog-
format=row
```


PERCONA

Any Complicated Setup

```
--let $rpl_topology= 1->2,2->3,3->1
--source include/rpl_init.inc
```


PERCONA

Any Complicated Setup

```
--let $rpl_topology= 1->2,2->3,3->1
--source include/rpl_init.inc

# On server 1
--let $rpl_connection_name= server_1
--source include/rpl_connection.inc
create table t1(id int) engine=innodb;
insert into t1 values(1);
```

Any Complicated Setup

```
--let $rpl_topology= 1->2,2->3,3->1
--source include/rpl_init.inc

# On server 1
--let $rpl_connection_name= server_1
--source include/rpl_connection.inc
create table t1(id int) engine=innodb;
insert into t1 values(1);

# On server 2
--let $rpl_connection_name= server_2
--source include/rpl_connection.inc
create table t2(id int) engine=innodb;
insert into t2 values(2);
```

Any Complicated Setup

```
# On server 3
--let $rpl_connection_name= server_3
--source include/rpl_connection.inc
create table t3(id int) engine=innodb;
insert into t3 values(3);
```


PERCONA

Complicated Replication Options

```
$ cat suite/mtr_test/t/rpl_mtr_test_9.cnf
!include ../../rpl/my.cnf
```

Complicated Replication Options

```
$ cat suite/mtr_test/t/rpl_mtr_test_9.cnf
!include ../../rpl/my.cnf

[mysqld.1]
log-slave-updates
gtid_mode=ON
enforce-gtid-consistency
```

PERCONA

Complicated Replication Options

```
$ cat suite/mtr_test/t/rpl_mtr_test_9.cnf
!include ../../rpl/my.cnf
[mysqld.2]
#master-info-repository=TABLE
#relay-log-info-repository=TABLE
log-slave-updates
gtid_mode=ON
enforce-gtid-consistency
[mysqld.3]
# Third server options
[ENV]
SERVER_MYPORT_3=        @mysqld.3.port
SERVER_MYSOCK_3=        @mysqld.3.socket
```


PERCONA

Run by Single Command

```
$ ./mtr --suite=your_suite your_test
Logging: ./mtr
MySQL Version
...
worker[1] Using MTR_BUILD_THREAD 300, with reserved ports 13000..13009
create table t1(f1 int);
insert into t1 values(1);
select * from t1;
f1
1
drop table t1;
your_suite.your_test                        [ pass ]  45802
...
Completed: All 1 tests were successful.
```


PERCONA

Run by Single Command

- Record result:

```
$ ./mtr --record --suite=your_suite your_test
Logging: ./mtr
...
your_suite.your_test                        [ pass ]  45802
------------------------------------------------------------------------
The servers were restarted 0 times
Spent 45.802 of 61 seconds executing testcases

Completed: All 1 tests were successful.
```

PERCONA

More Information


www.percona.com/downloads/


dev.mysql.com/downloads/


Installing from generic binaries


Test databases


DBdeployer - Beto - Percona


DBdeployer - ProxySQL


anydbver


MySQL Test Framework

PERCONA


PERCONA®
/svetsmirnova
/svetasmirnova
/svetsmirnova
/SvetaSmirnova