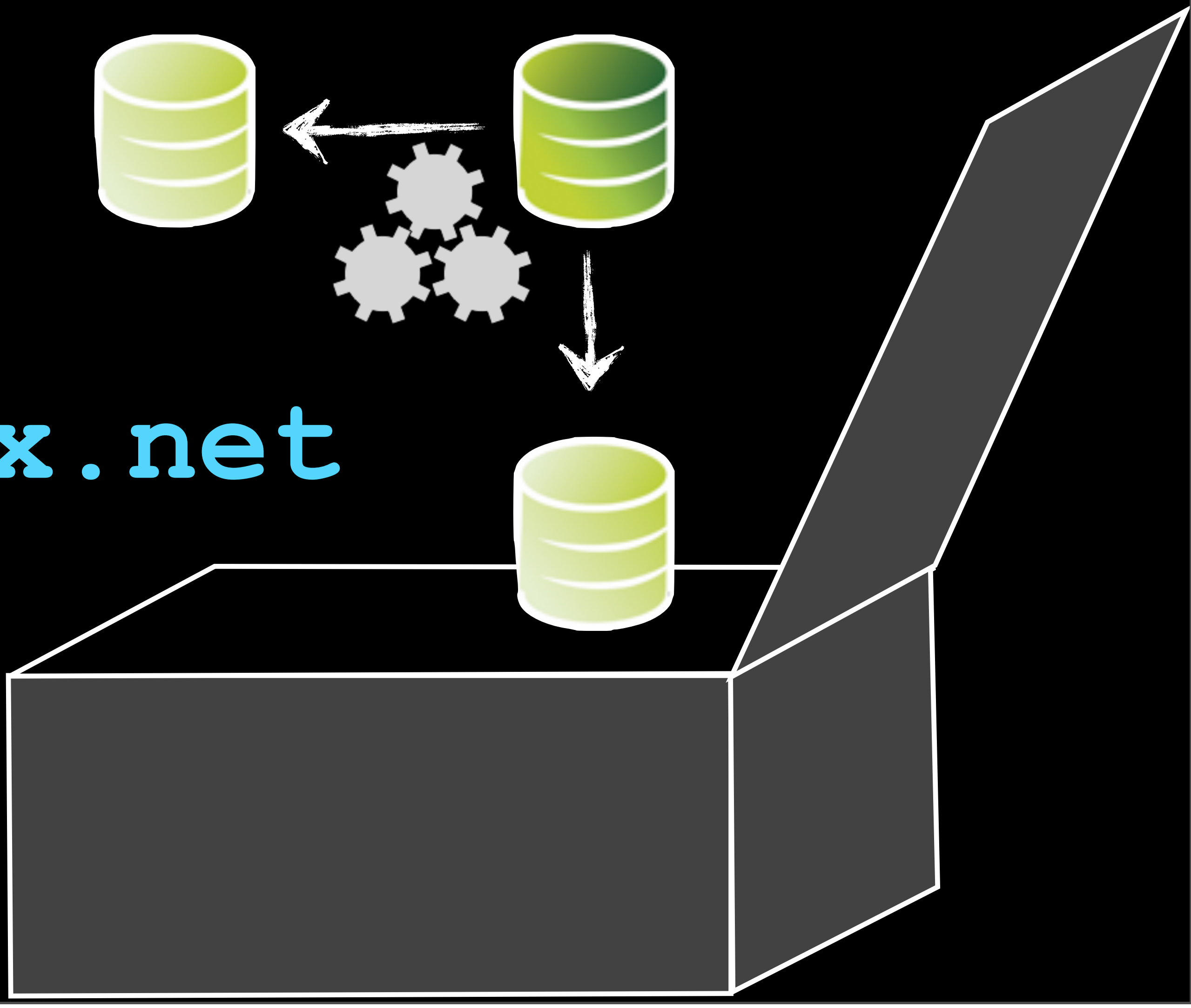
Testing MySQL creatively in a Sandbox

Giuseppe Maxia
QA Director, Continuent, Inc

http://mysqlsandbox.net



about me - Giuseppe Maxia

- a.k.a. The Data Charmer
- QA Director at Continuent, Inc
- Long time hacking with MySQL features
- Formerly, community manager,db consultant, designer, coder.
- A passion for QA and open source
- Blogger
- http://datacharmer.blogspot.com

MySQL Sandbox lightning presentation

I used to install a lot of MySQL databases for testing

MANUALLY

Then, I decided to use Perl ...

DBA pop quiz

HOW MANY KEYSTROKES to install a MySQL server?

HOW MANY KEYSTROKES to install a MySQL server?

10

```
sb 5.0.83↵
1234567890
```

HOW MANY KEYSTROKES to install 3 MySQL servers in replication?

HOW MANY KEYSTROKES to install 3 MySQL servers in replication?

```
         11
sb r5.0.83↵
12345678901
```

HOW LONG
does it take
to install a MySQL server?

HOW LONG
does it take
to install a MySQL server?

< 5 seconds

```
time sb 5.0.83
0m1.518s
```

HOW LONG does it take to install 3 MySQL servers in replication?

HOW LONG
does it take
to install 3 MySQL servers in replication?

< 10 seconds

```
sb 5.0.83
```

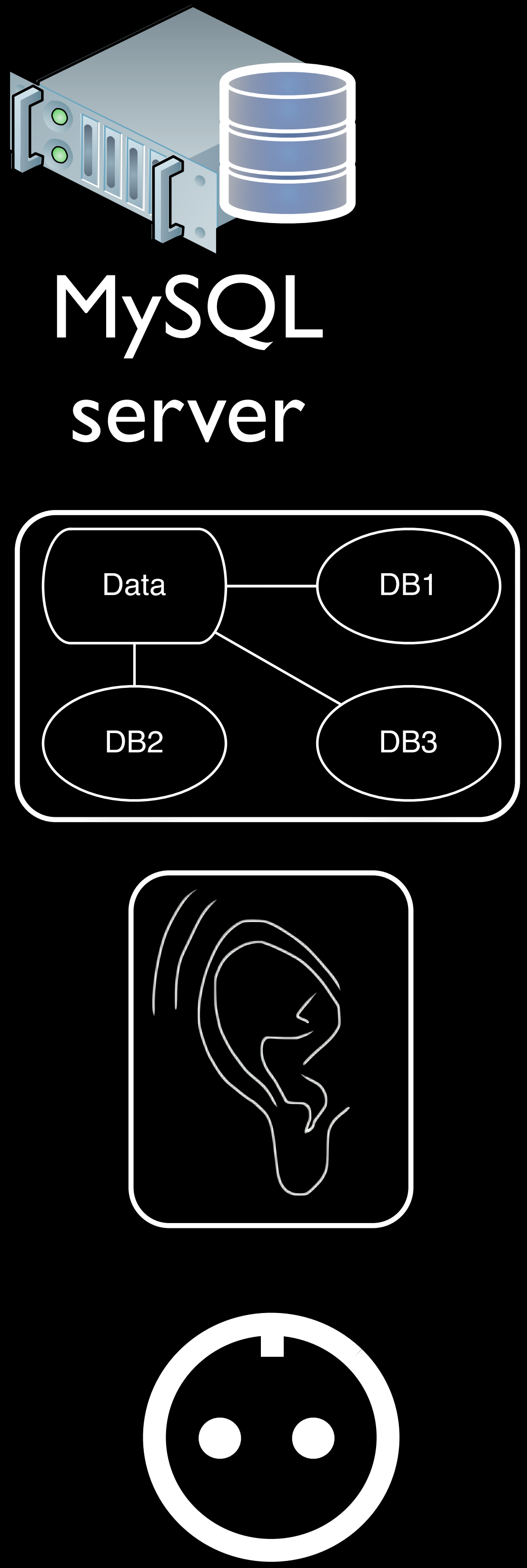
```
0m4.515s
```

MySQL Sandbox

http://mysqlsandbox.net

- Free software (Perl under GPL)
- One (unix) host
- Many database servers
- Single or multiple sandboxes
- Customized scripts to use the servers
- Standard or circular replication
- Installs **IN SECONDS**

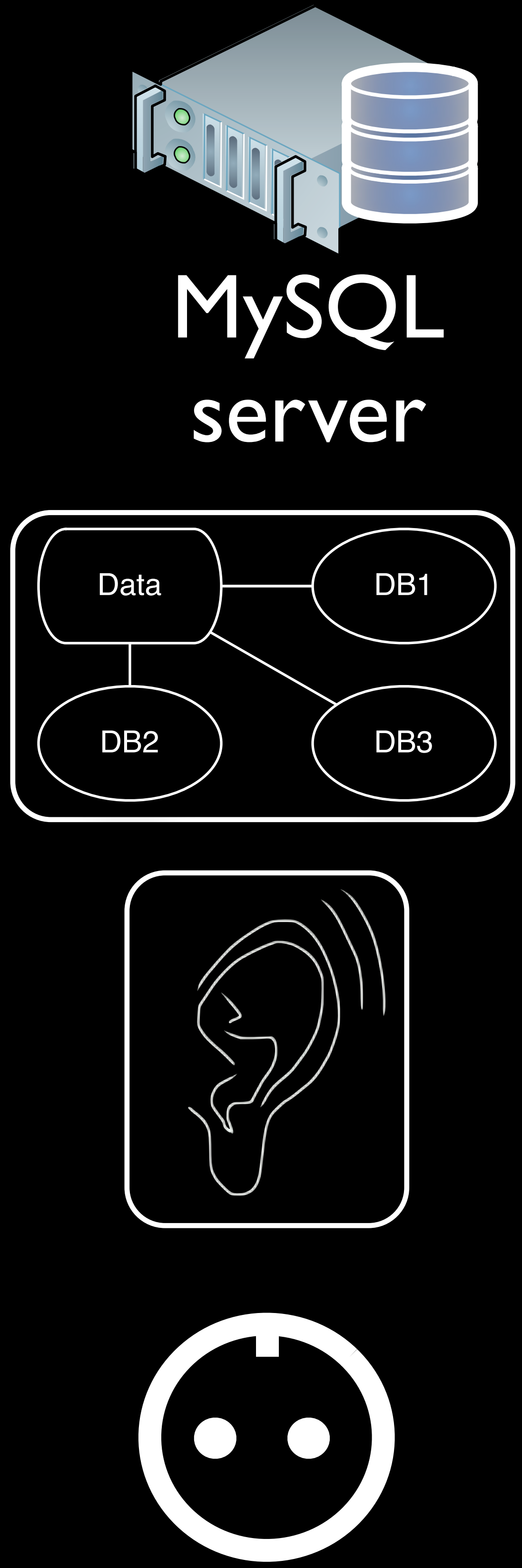
overview



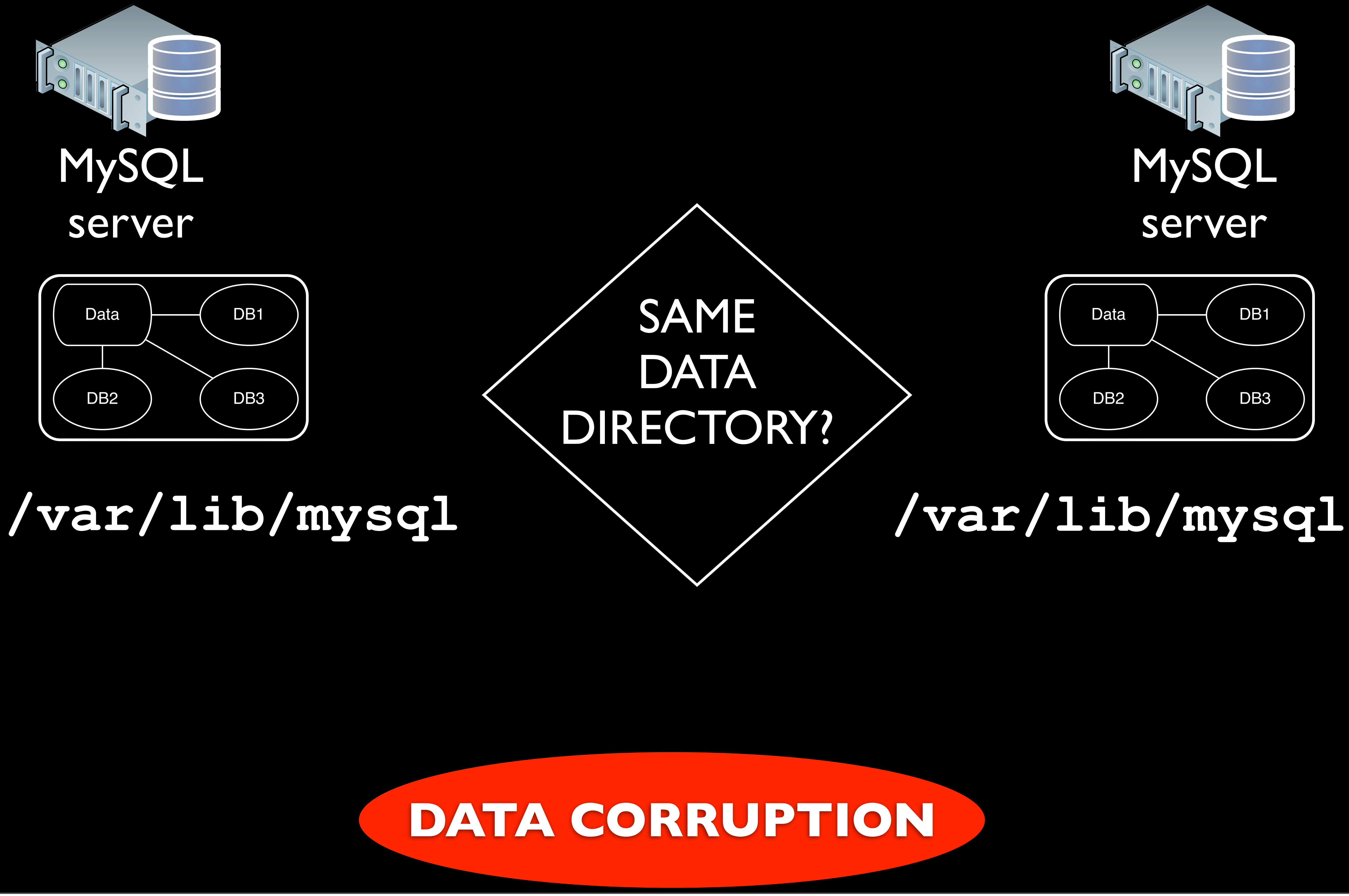
DATA DIRECTORY

PORT

SOCKET



overview



MySQL server

MySQL server

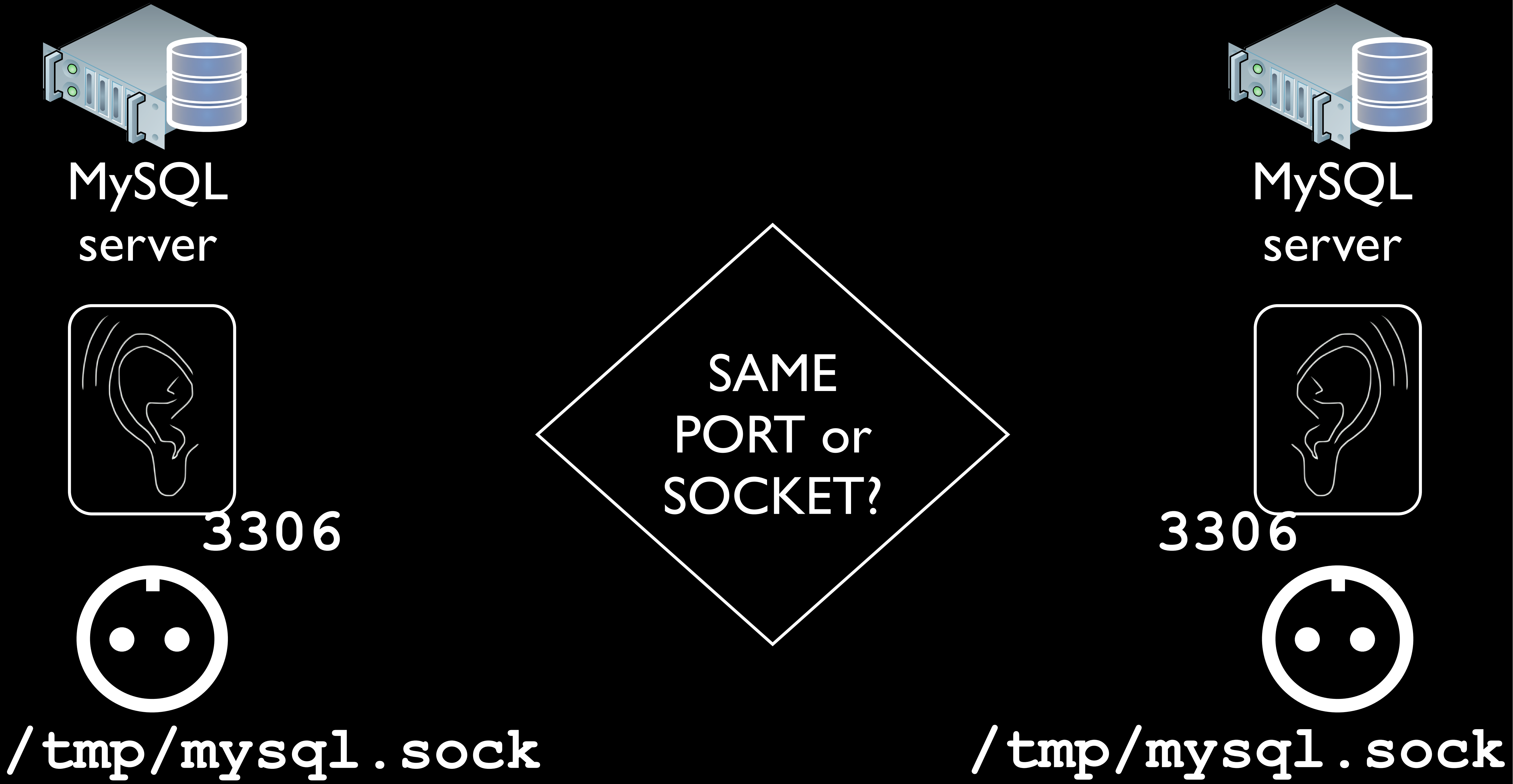
/var/lib/mysql

SAME DATA DIRECTORY?

/var/lib/mysql

DATA CORRUPTION

overview



MySQL server

3306

/tmp/mysql.sock

SAME PORT or SOCKET?

MySQL server

3306

/tmp/mysql.sock

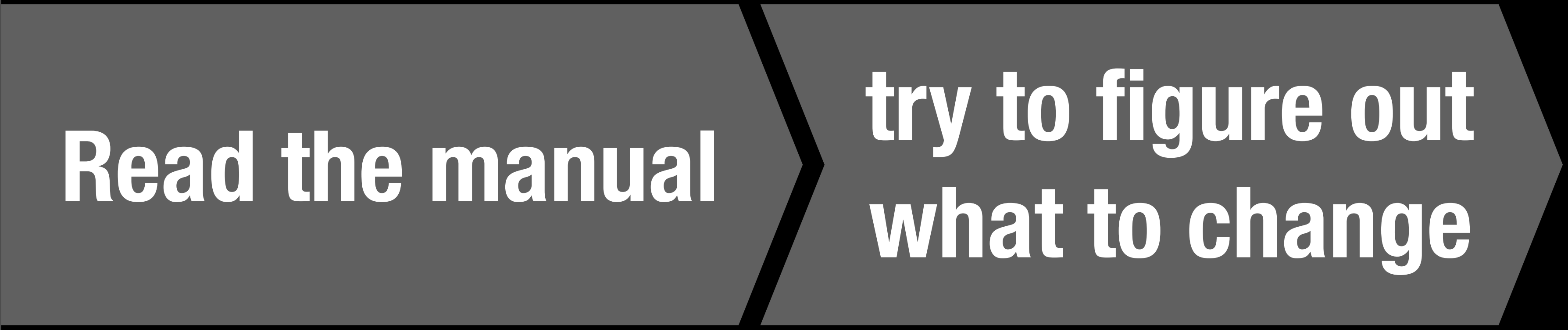
DOES NOT START

The hard way

The hard way

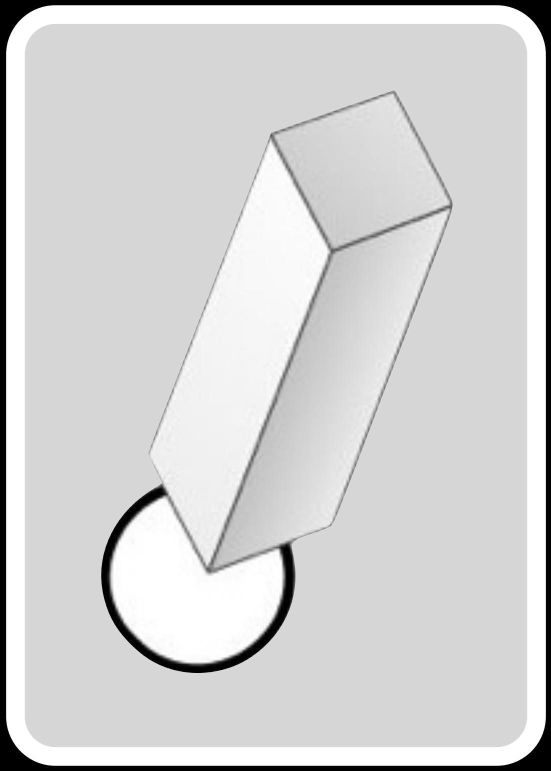
Read the manual

The hard way

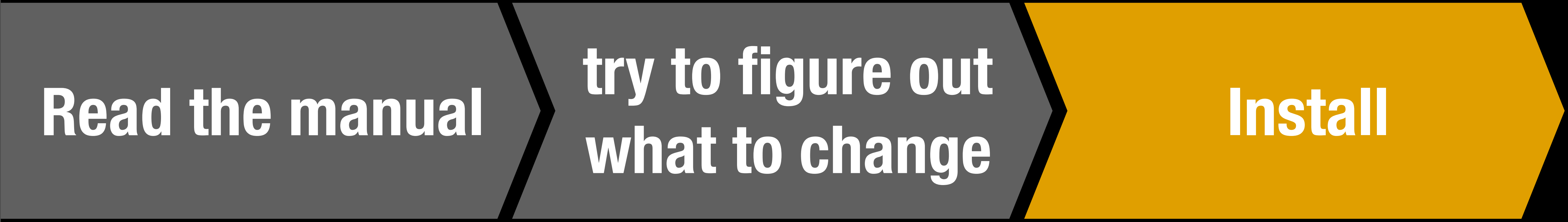


Read the manual

try to figure out what to change



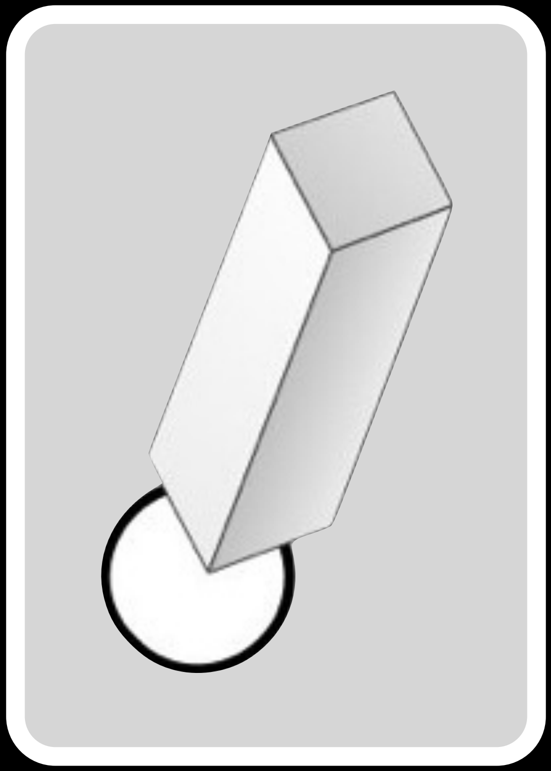
The hard way



Read the manual

try to figure out what to change

Install

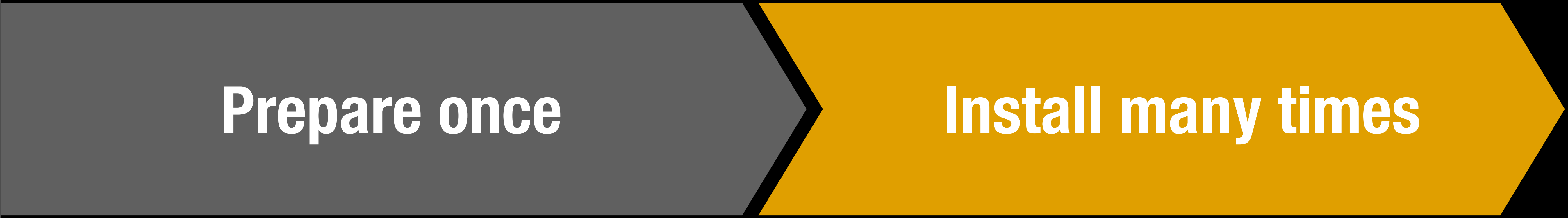


The easy way

MySQL Sandbox

```
$ make_sandbox \
    /path/to/mysql-5.1.54_linux.tar.gz

# it should work always
```

The easier way



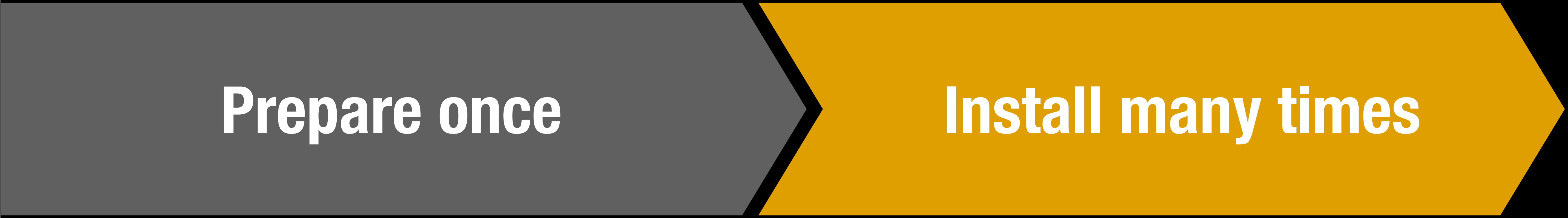
Prepare once

Install many times

```
# some
# preliminary
# work
```

```
$ make_sandbox 5.1.54
```

The easiest way



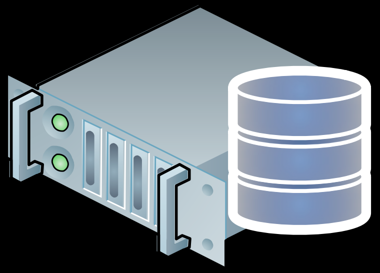
Prepare once

Install many times

```
# some
# preliminary
# work
```

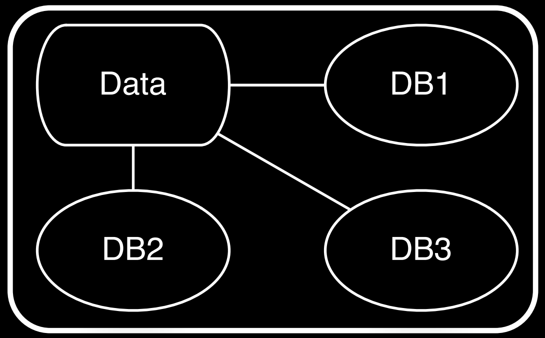
```
$ sb 5.1.54
```

MySQL Sandbox



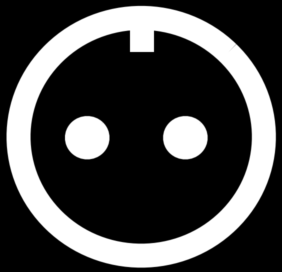
MySQL
server

VERSION



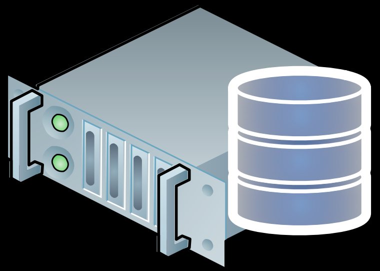
```
$SANDBOX_HOME/msb_VERSION/data
```

VERSION



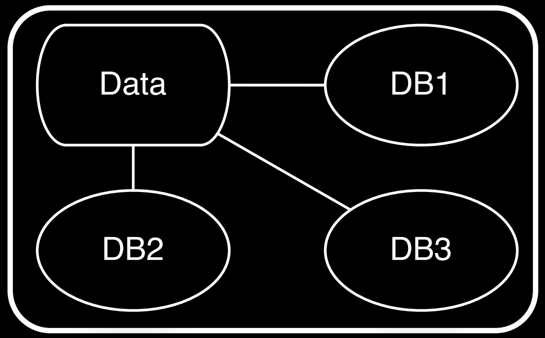
```
/tmp/mysql_VERSION.sock
```

MySQL Sandbox



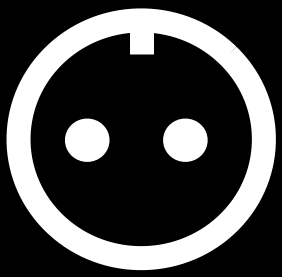
MySQL
server

5.1.54



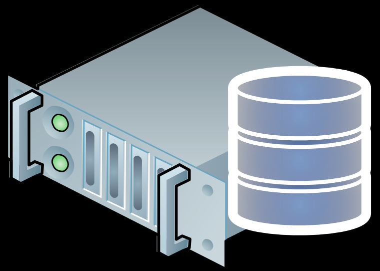
$SANDBOX_HOME/msb_5_1_54/data

5154



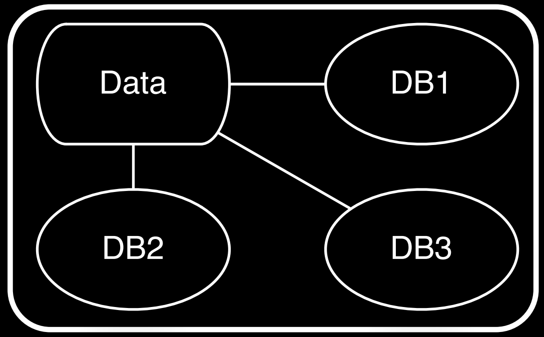
/tmp/mysql_5154.sock

MySQL Sandbox



MySQL
server

`5.5.9`



`$SANDBOX_HOME/msb_5_5_09/data`

`5509`

`/tmp/mysql_5509.sock`

Single Sandbox

MySQL server

customized scripts

```
start
stop
restart
status
clear
send_kill
use
```

Multiple Sandbox

MySQL
server

customized scripts

```
start_all
stop_all
restart_all
status_all
clear_all
send_kill_all
use_all
```

```
m
s1
s2
```

```
n1
n2
n3
```

Where do you get it

- from CPAN

```
sudo su -
cpan MySQL::Sandbox
```

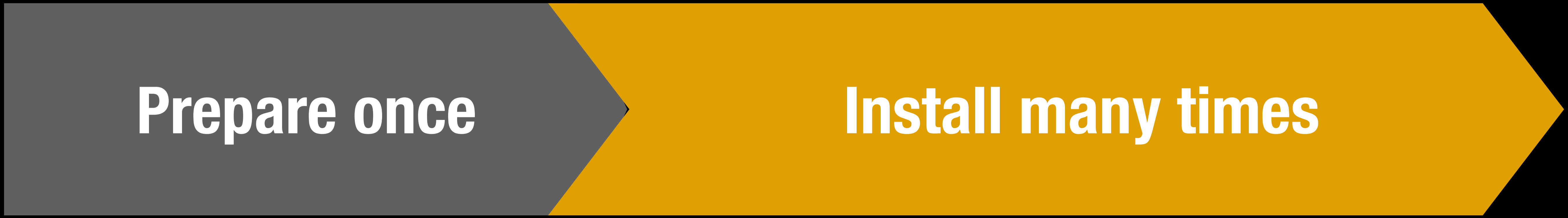
- from launchpad

```
http://launchpad.net/mysql-sandbox
```

The easy replication way

MySQL Sandbox

```
$ make_replication_sandbox \
    /path/to/mysql-5.1.54_linux.tar.gz
```



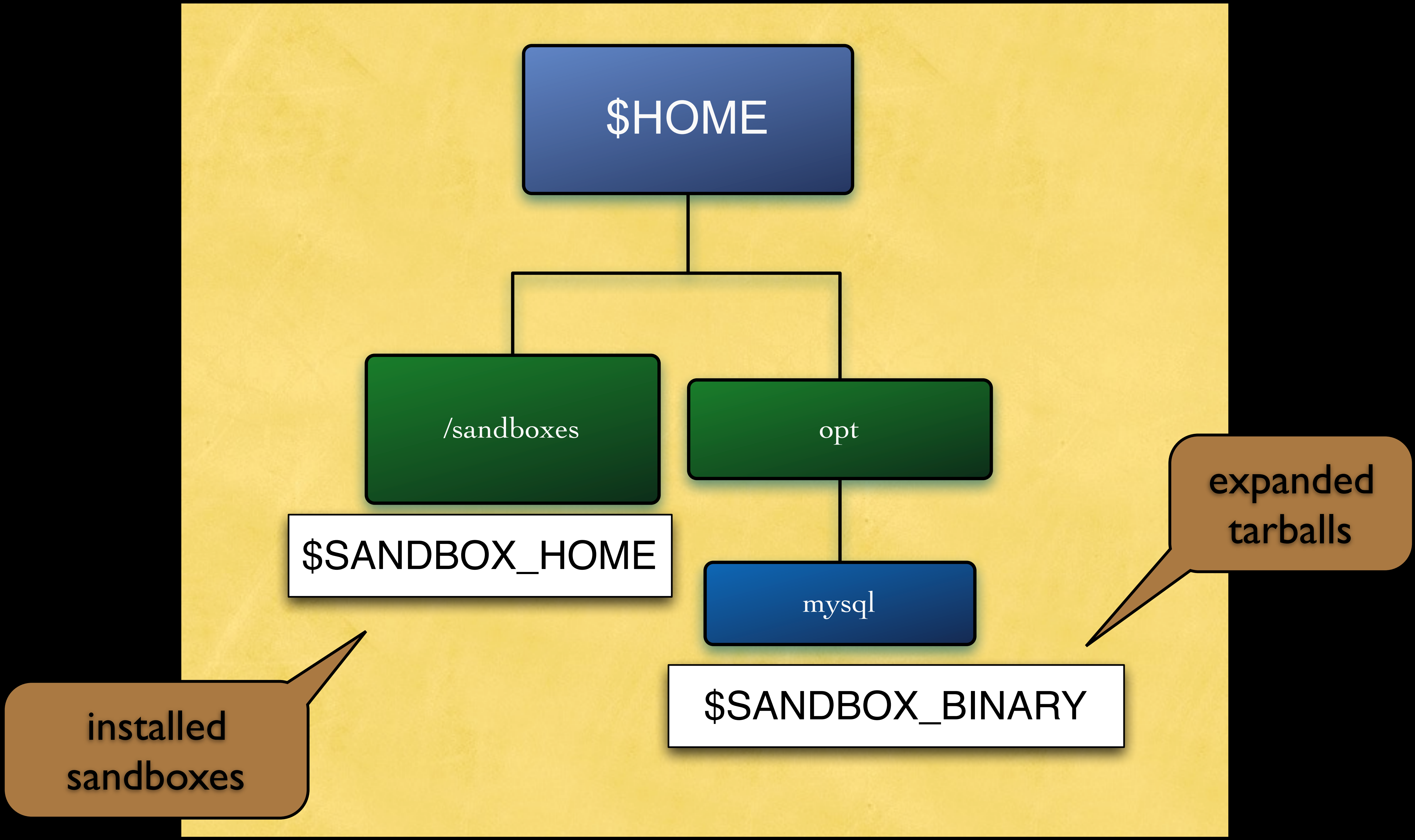
Prepare once

Install many times

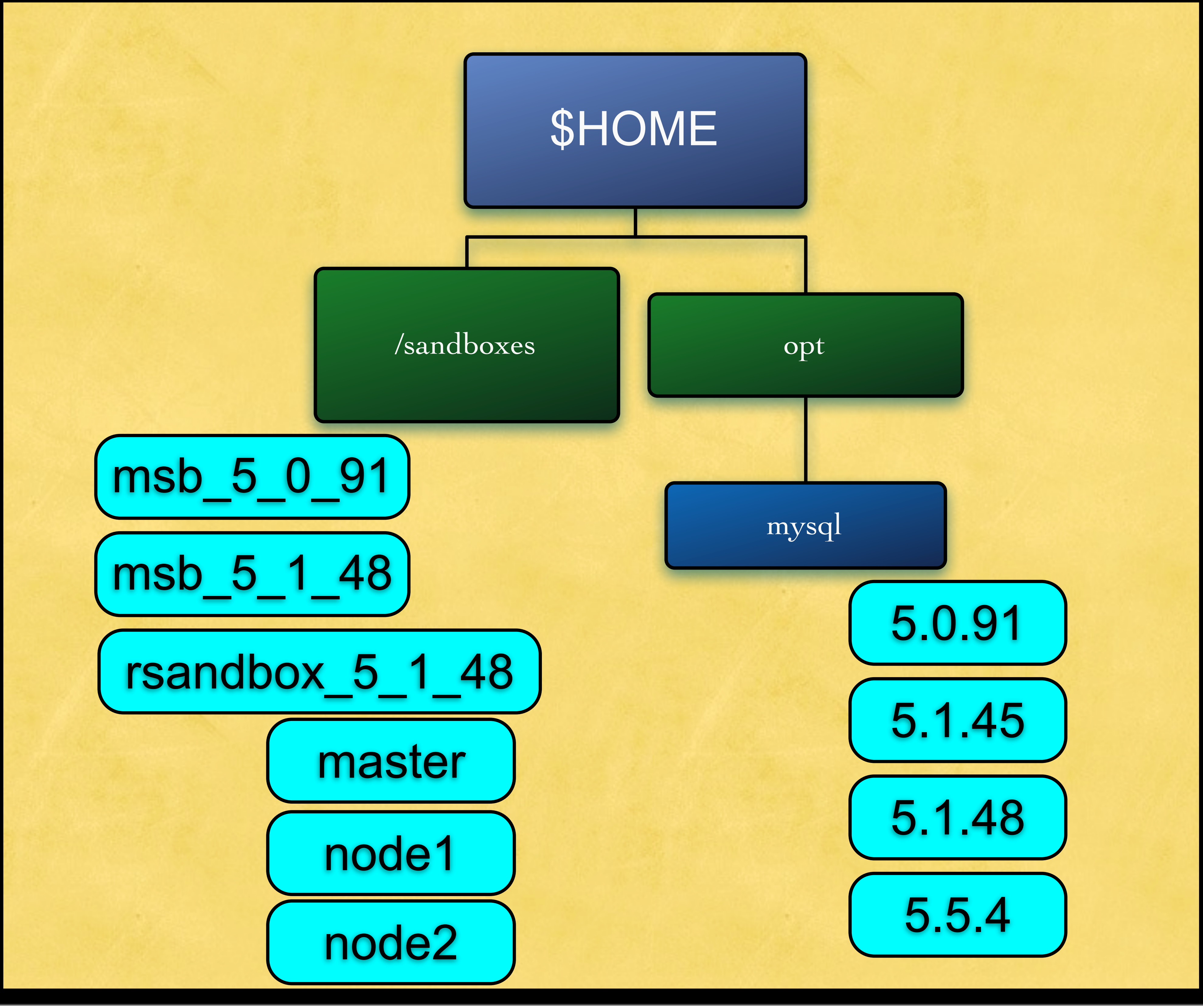
```
# some
# preparation
```

```
$ make_replication_sandbox
5.1.54
```

default architecture



default architecture



COOKBOOK

creating a single sandbox

```
make_sandbox \
   /path/to/mysql-X.X.XX-OS.tar.gz
```

using a single sandbox

```
# after
# make_sandbox \
#   /path/to/mysql-X.X.XX-OS.tar.gz

$ cd $SANDBOX_HOME/msb_X_X_XX
$ ./use
```

creating a single sandbox with a specific options file

```
make_sandbox \
   /path/to/mysql-X.X.XX-OS.tar.gz \
   --my_file=/path/to/my.cnf
```

easily create a sandbox after the first one

The long way

```
$ cd $HOME/opt/mysql    # $SANDBOX_BINARY
$   gunzip -c \
  /path/to/mysql-5.1.34-osx10.5-x86.tar.gz \
    | tar -xf -
$ mv mysql-5.1.34-osx10.5-x86 5.1.34
$ make sandbox 5.1.34
```

easily create a sandbox after the first one

The short way

```
$ make_sandbox --export_binaries \
  path/to/mysql-5.1.34-osx10.5-x86.tar.gz
```

starting a single sandbox

```
$ cd $SANDBOX_HOME/msb_X_X_XX
$ ./start
```

starting a single sandbox with temporary options

```
$ cd $SANDBOX_HOME/msb_X_X_XX
$ ./start --option=value
```

```
$ ./restart --option=value
```

```
$ ./start --key-buffer=20000000
```

creating a sandbox with custom port and directory

```
$ make_sandbox 5.1.34 -- \
   --sandbox_port=7800 \
   --sandbox_directory=mickeymouse
```

creating a sandbox with automatic port checking

```
$ make_sandbox 5.1.34 -- --check_port


# if 5.1.34 is free
#    port=5134
#    directory=msb_5_1_34
# else
#    port=5135 (or the first free)
#    directory=msb_5_1_34_a
```

create a replication sandbox

```
$ make_replication_sandbox \
  path/to/mysql-5.1.34-osx10.5-x86.tar.gz
```

create a circular replication sandbox

```
$ make_replication_sandbox \
  --circular=4 \
  path/to/mysql-5.1.34-osx10.5-x86.tar.gz
```

changing port to an existing sandbox

```
$ sbtool -o port \
    -s /path/to/source/sandbox \
    --new_port=XXXX
```

installing the innodb plugin

```
$ sbtool -o plugin \
    --plugin=innodb \
    -s /path/to/source/sandbox
```

creating a replication sandbox with new base port

```
$ make_replication_sandbox \
    --replication_directory=newwdir \
    --check_base_port 5.0.79

# Creates a replication directory under
# $SANDBOX_HOME/newdir
# The previous one is preserved.
# No conflicts happen
```

creating a stand-alone master

```
$ make_sandbox 5.5.16 -- --master

# Creates a sandbox with binary log and
# server-id enabled
```

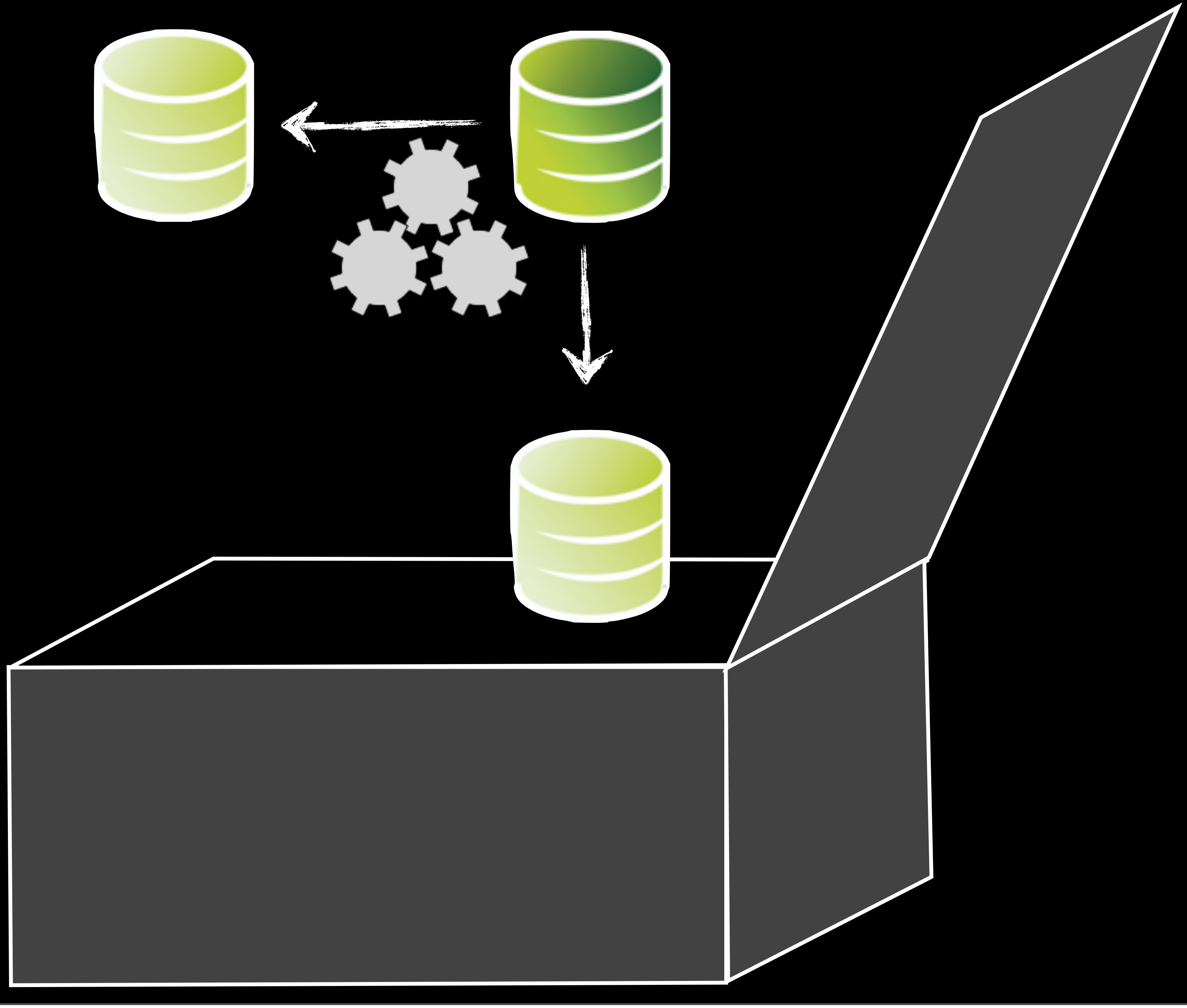
creating a quick slave

```
~/sandboxes/rsandbox_5_5_16/m -e 'show
variables like "port"'
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| port          | 19771 |
+---------------+-------+

$ make_sandbox 5.5.16 -- \
  --slaveof='master_port=19771'

# adds a slave to an existing master
```

MySQL Sandbox cookbook

```
$ perldoc MySQL::Sandbox::Recipes
```

testing MySQL creatively



Breaking replication

```
make_replication_sandbox 5.5.16

cd $HOME/sandboxes/rsandbox_5_5_16

./m -e 'create table t1 (i int not null primary key)'
test
./s1 -e 'insert into t1 values (1)' test
./m -e 'insert into t1 values (1)' test
./s1 -e 'show slave status\G' | grep 'Error\|Running'
             Slave_IO_Running: Yes
             Slave_SQL_Running: No
                    Last_Error: Error 'Duplicate entry '1' for key
'PRIMARY'' on query. Default database: 'test'. Query: 'insert into t1
values (1)'
                 Last_IO_Error:
                Last_SQL_Error: Error 'Duplicate entry '1' for key
'PRIMARY'' on query. Default database: 'test'. Query: 'insert into t1
values (1)'
```

Fixing replication

```
./s1 -e 'delete from t1 where i = 1' test
./s1 -e 'start slave'
./s1 -e 'show slave status\G' | grep 'Error\|Running'
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
                   Last_Error:
                Last_IO_Error:
               Last_SQL_Error:
```

Making a slave lag

```
./s1 -e 'stop slave SQL_THREAD'
./s2 -e 'stop slave SQL_THREAD'

./m < heavy_load_commands.sql


# 1 hour later

./s1 -e 'start slave SQL_THREAD'
./s2 -e 'start slave SQL_THREAD'
```

Options for replication nodes

```
--master_options = name      Options passed to the master
--slave_options = name       Options passed to each slave
--node_options = name        Options passed to each node
--one_slave_options = name   Options passed to a specific slave
                              with the format "N:options"
```

customizing sandboxes during installation (1)

```
make_replication_sandbox \
   --node_options=--high_performance \
   --one_slave_options='1:-c read-only'\
   5.1.57
```

customizing sandboxes during installation (2)

```
./use_all 'show variables like "%innodb%buffer%"'
# master
Variable_name Value
innodb_buffer_pool_size  536870912
innodb_log_buffer_size 52428800
# server: 1:
Variable_name Value
innodb_buffer_pool_size  536870912
innodb_log_buffer_size 52428800
# server: 2:
Variable_name Value
innodb_buffer_pool_size  536870912
innodb_log_buffer_size 52428800
```

customizing sandboxes during installation (3)

```
./use_all 'show variables like
"%read_only%"'
# master
Variable_name Value
read_only OFF
# server: 1:
Variable_name Value
read_only ON
# server: 2:
Variable_name Value
read_only OFF
```

re-playing binary logs

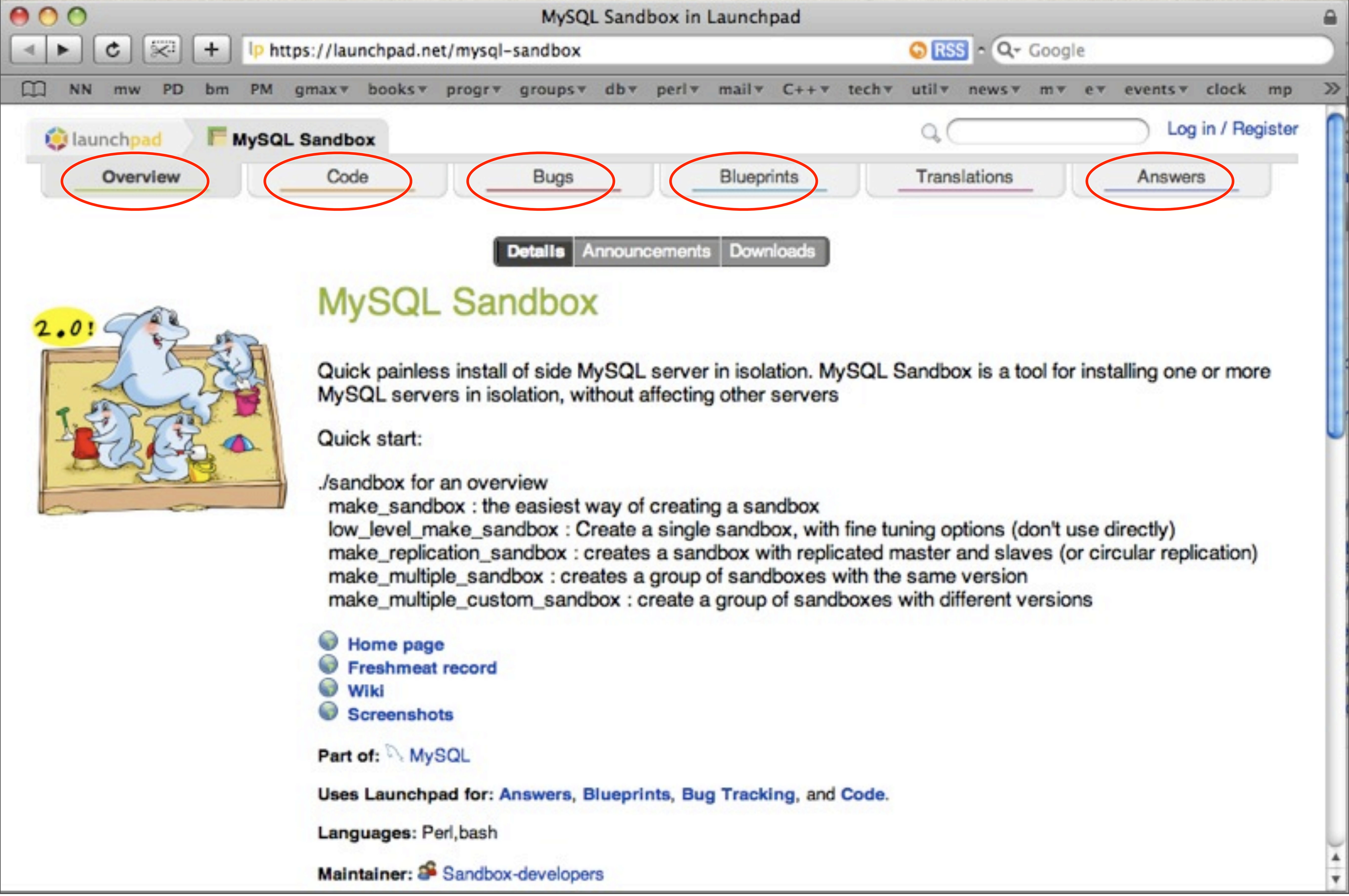
- 1) server become unusable for human error
- 2) install a new server
- 3) load the latest backup
- 4) create a sandbox and copy the binary logs to its data directory
- 5) make the new server a slave of the sandbox
- 6) wait for completion
- 7) reset slave
- 8) remove the sandbox

Testing Percona Server

```
make_sandbox \
  --add_prefix=ps \
  --export_binaries \
  Percona-Server-5.5.21-osx10.7-.tar.gz

# and later

make_sandbox ps5.5.21
```

DEMO

Participate!


MySQL Sandbox in Launchpad
https://launchpad.net/mysql-sandbox
RSS
Google
NN mw PD bm PM gmax books progr groups db perl mail C++ tech util news m e events clock mp
launchpad
MySQL Sandbox
Log in / Register
Overview
Code
Bugs
Blueprints
Translations
Answers
Details
Announcements
Downloads
2.0!

MySQL Sandbox

Quick painless install of side MySQL server in isolation. MySQL Sandbox is a tool for installing one or more MySQL servers in isolation, without affecting other servers

Quick start:

./sandbox for an overview
  make_sandbox : the easiest way of creating a sandbox
  low_level_make_sandbox : Create a single sandbox, with fine tuning options (don't use directly)
  make_replication_sandbox : creates a sandbox with replicated master and slaves (or circular replication)
  make_multiple_sandbox : creates a group of sandboxes with the same version
  make_multiple_custom_sandbox : create a group of sandboxes with different versions

- Home page
- Freshmeat record
- Wiki
- Screenshots

Part of: MySQL

Uses Launchpad for: Answers, Blueprints, Bug Tracking, and Code.

Languages: Perl,bash

Maintainer: Sandbox-developers

THANKS

http://mysqlsandbox.net