

# MySQL Sandbox

- one (unix) host
- many database servers
- single or multiple sandboxes
- installs **IN SECONDS**
- Free software (GPL)

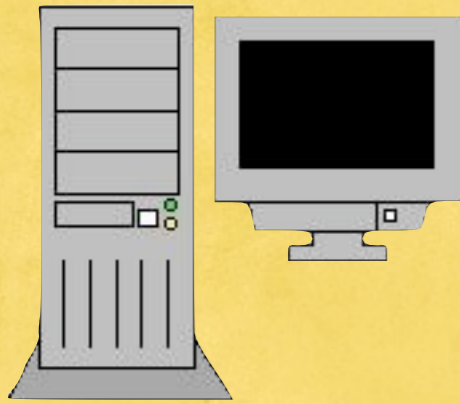


<http://mysqlsandbox.net>

Monday, 20 July 2009

MySQL Sandbox is a tool that installs an additional MySQL server in a few seconds. You can have more than one database servers in your host, all of them independent from each other. Installing a new server with MySQL Sandbox is a matter of seconds. You can install either a single sandbox or a system including multiple ones. It is free software, and it works on most every unix system.

# 1 host - many servers



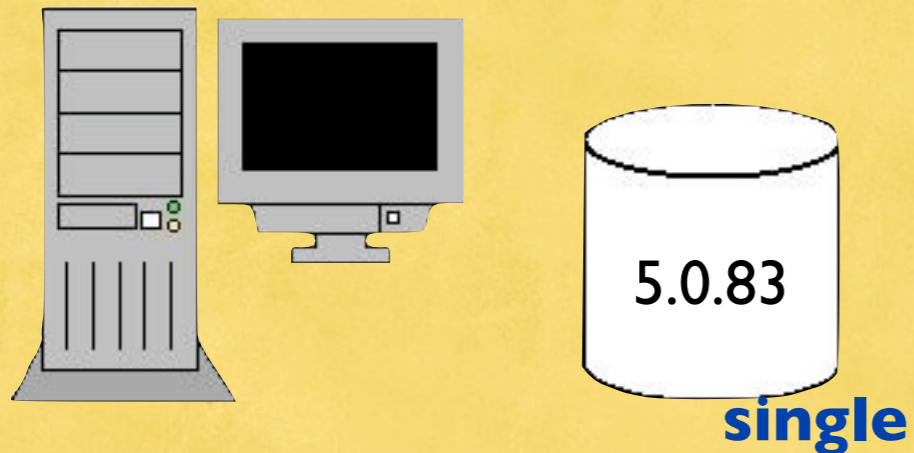
HOST



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

# 1 host - many servers



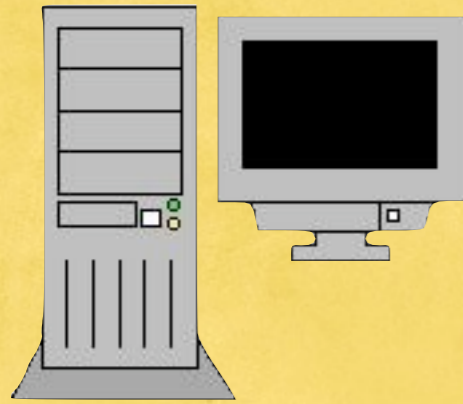
HOST



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

# 1 host - many servers



single



single

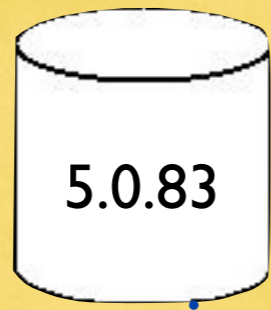
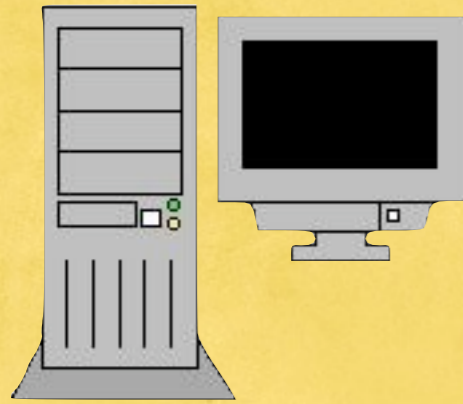
HOST



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

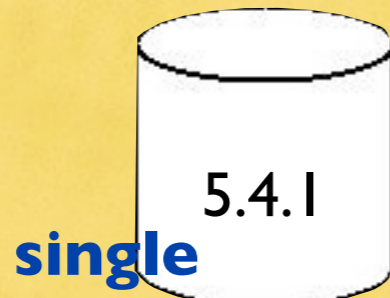
# 1 host - many servers



single



single



single

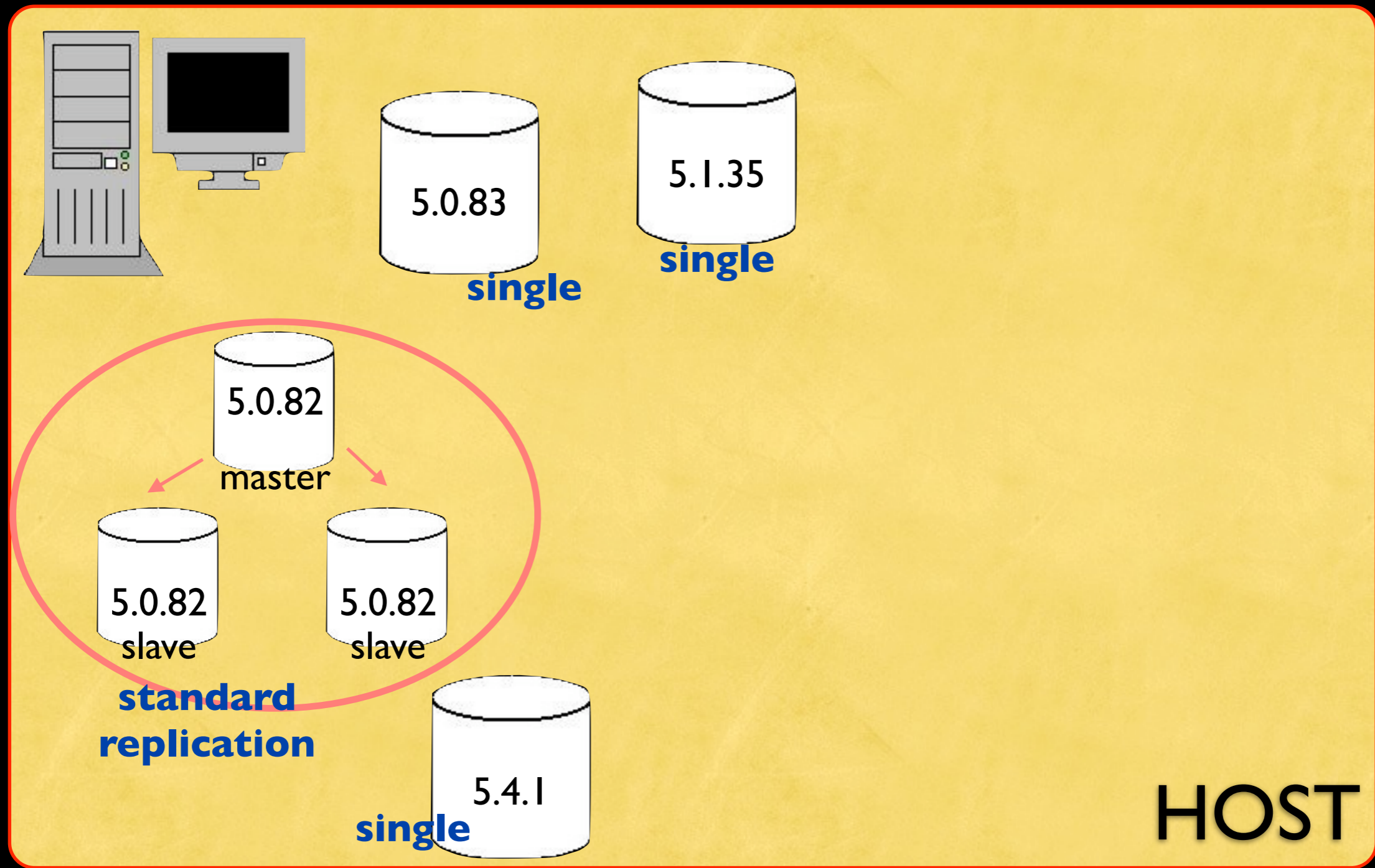
HOST



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

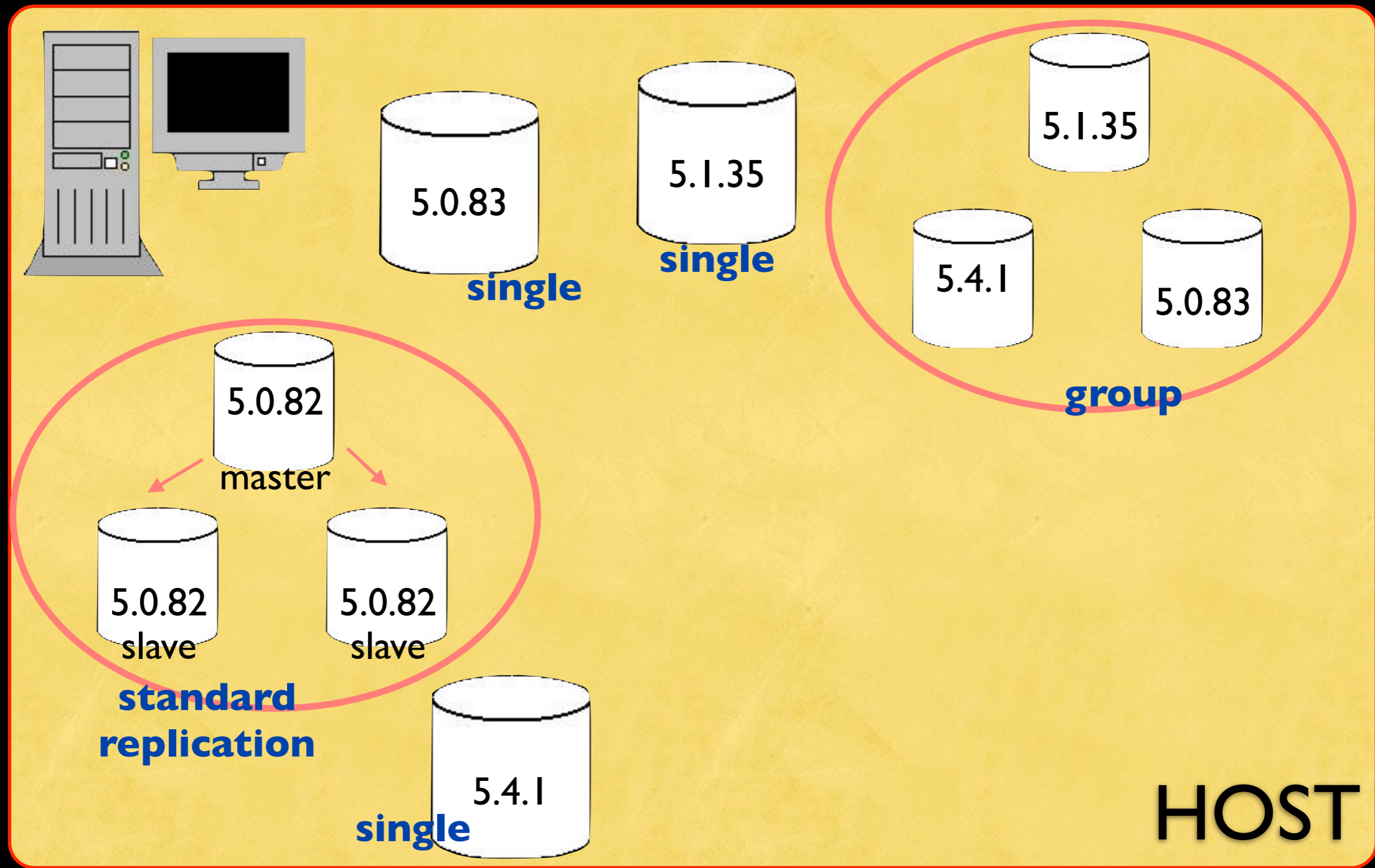
# 1 host - many servers



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

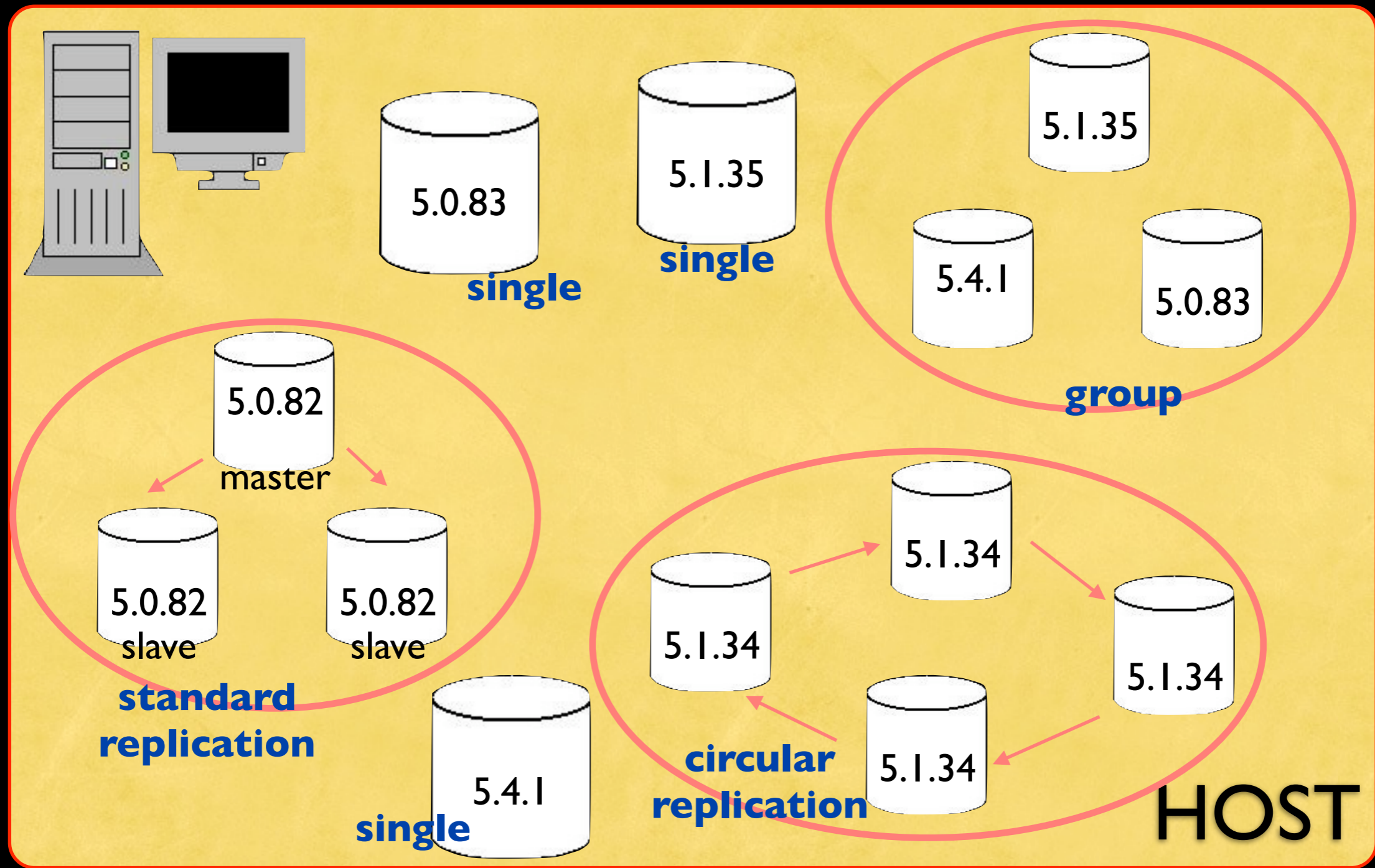
# 1 host - many servers



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

# 1 host - many servers

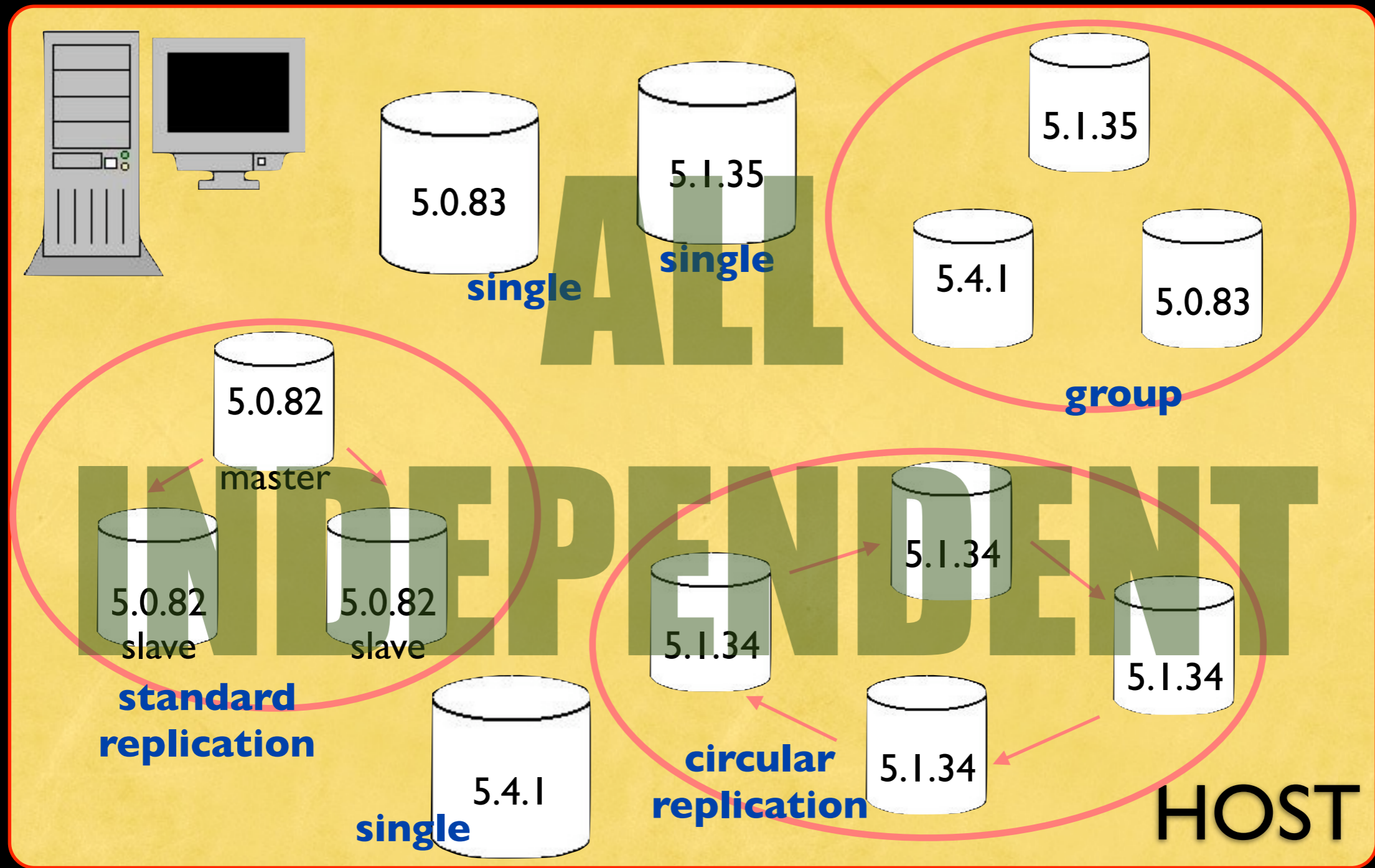


Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.



# 1 host - many servers



Monday, 20 July 2009

When you need to test enterprise software, the database is a core component. For this, it's important to be able to have test database servers in your host. You want to be able to test in different conditions, possibly with different versions. With MySQL Sandbox, it's easy to create single servers of different versions, or a replicated system with master and slaves, or a group of servers with various versions, or a circular replication system. You can have all of them, one by one, or all at once, and they will not interfere with each other.

# Installation

- as root

```
# cpan MySQL::Sandbox
```



Monday, 20 July 2009

Installing MySQL Sandbox is easy. It is a Perl module. Getting it in your server is just a matter of typing one command as root. If you want, you can also install in the user space. The reference manual explains how to do it.

# Creating a single sandbox

- as normal user

```
# make_sandbox \
```

```
/path/to/mysql-5.1.35-YOUROS.gz
```



Monday, 20 July 2009

Let's make your first sandbox. It's as simple as saying "make\_sandbox" and the name of a MySQL tarball.

This will expand the tarball and create a sandbox very fast.

If you have already expanded the tarball in a predefined directory, you can call "make\_sandbox" with the simplified syntax.

It is FAST!

# Creating a single sandbox

- as normal user

```
# make_sandbox \
```

```
  /path/to/mysql-5.1.35-YOUROS.gz
```

- if you have expanded tarballs already



Monday, 20 July 2009

Let's make your first sandbox. It's as simple as saying "make\_sandbox" and the name of a MySQL tarball.

This will expand the tarball and create a sandbox very fast.

If you have already expanded the tarball in a predefined directory, you can call "make\_sandbox" with the simplified syntax.

It is FAST!

# Creating a single sandbox

- as normal user

```
# make_sandbox \
```

```
  /path/to/mysql-5.1.35-YOUROS.gz
```

- if you have expanded tarballs already

```
# make_sandbox 5.1.35
```



# Creating a single sandbox

- as normal user

```
# make_sandbox \
```

```
/path/to/mysql-5.1.35-YOUROS.gz
```

< **10 seconds!**

- if you have expanded tarballs already

```
# make_sandbox 5.1.35
```



Monday, 20 July 2009

Let's make your first sandbox. It's as simple as saying "make\_sandbox" and the name of a MySQL tarball. This will expand the tarball and create a sandbox very fast.

If you have already expanded the tarball in a predefined directory, you can call "make\_sandbox" with the simplified syntax. It is FAST!

# Creating a replication sandbox

- as normal user

```
# make_replication_sandbox \  
/path/to/mysql-5.1.35-YOUROS.gz
```



Monday, 20 July 2009

Creating a replication system is equally easy. If you have ever set up MySQL replication servers you know that it is not extremely complicated, but the chances of making mistakes are quite a lot.

Using MySQL Sandbox, you can create a replication system as easily as you make a single one. And you will be surprised at how fast it is!

# Creating a replication sandbox

- as normal user

```
# make_replication_sandbox \  
/path/to/mysql-5.1.35-YOUROS.gz
```

- if you have expanded tarballs already



Monday, 20 July 2009

Creating a replication system is equally easy. If you have ever set up MySQL replication servers you know that it is not extremely complicated, but the chances of making mistakes are quite a lot.

Using MySQL Sandbox, you can create a replication system as easily as you make a single one. And you will be surprised at how fast it is!



# Creating a replication sandbox

- as normal user

```
# make_replication_sandbox \  
    /path/to/mysql-5.1.35-YOUROS.gz
```

- if you have expanded tarballs already

```
# make_replication_sandbox 5.1.35
```



Monday, 20 July 2009

Creating a replication system is equally easy. If you have ever set up MySQL replication servers you know that it is not extremely complicated, but the chances of making mistakes are quite a lot.

Using MySQL Sandbox, you can create a replication system as easily as you make a single one. And you will be surprised at how fast it is!

# Creating a replication sandbox

- as normal user

```
# make_replication_sandbox \
```

```
/path/to/mysql-5.1.35-YOUROS.gz
```

< **20 seconds!**

- if you have expanded tarballs already

```
# make_replication_sandbox 5.1.35
```



Monday, 20 July 2009

Creating a replication system is equally easy. If you have ever set up MySQL replication servers you know that it is not extremely complicated, but the chances of making mistakes are quite a lot.

Using MySQL Sandbox, you can create a replication system as easily as you make a single one. And you will be surprised at how fast it is!

# SHORTCUT - Creating and Using a sandbox



Monday, 20 July 2009

Starting with MySQL Sandbox 3.0.04, there is also a shortcut script that can make your life much easier. You can create a sandbox and use it with a very short command. You need to try it to believe it.

# SHORTCUT - Creating and Using a sandbox

- as normal user (requires version 3.0.04)



Monday, 20 July 2009

Starting with MySQL Sandbox 3.0.04, there is also a shortcut script that can make your life much easier. You can create a sandbox and use it with a very short command. You need to try it to believe it.

# SHORTCUT - Creating and Using a sandbox

- as normal user (requires version 3.0.04)

**# sb 5.0.83**



Monday, 20 July 2009

Starting with MySQL Sandbox 3.0.04, there is also a shortcut script that can make your life much easier. You can create a sandbox and use it with a very short command. You need to try it to believe it.

# SHORTCUT - Creating and Using a sandbox

- as normal user (requires version 3.0.04)

**# sb 5.0.83**

**< 5 seconds!**



Monday, 20 July 2009

Starting with MySQL Sandbox 3.0.04, there is also a shortcut script that can make your life much easier. You can create a sandbox and use it with a very short command. You need to try it to believe it.

# example

```
$ make_sandbox 5.0.82
```

```
[...]
```

```
. sandbox server started
```

```
Your sandbox server was installed in  
$SANDBOX_HOME/msb_5_0_82
```

```
$ $SANDBOX_HOME/msb_5_0_82/use \  
-e "select version()"
```

```
+-----+  
| version() |  
+-----+  
| 5.0.82 |  
+-----+
```

# example (shortcut)

```
$ time sb 5.0.82 -e "select version()"
```

```
[...]
```

```
. sandbox server started
```

```
Your sandbox server was installed in  
$SANDBOX_HOME/msb_5_0_82
```

```
+-----+  
| version() |  
+-----+  
| 5.0.82    |  
+-----+
```

```
real    0m2.041s
```

```
user    0m0.290s
```

```
sys     0m0.210s
```



# example replication

```
$ make_replication_sandbox 5.0.82
```

```
[...]
```

```
replication directory installed on  
$SANDBOX_HOME/rsandbox_5_0_82
```

```
$ $SANDBOX_HOME/rsandbox_5_8_82/m \  
-e "select version()"
```

```
+-----+  
| version() |  
+-----+  
| 5.0.82    |  
+-----+
```

Monday, 20 July 2009

Creating a replication system is similar. Just use the `make_replication_sandbox` script instead of `make_sandbox`, and the program will take care of all the details.

Inside a replication sandbox you will have one sandbox for the master and one for each slave. No need to remember which is which. The sandbox offers a shortcut script for each role. "m" is for master, "s1" for the first slave, and so on.

# example replication (shortcut)

```
$ time sb r5.0.82 -e "select version()"
```

```
[...]
```

```
replication directory installed in $HOME/  
rsandbox_5_0_82
```

```
+-----+
```

```
| version() |
```

```
+-----+
```

```
| 5.0.82-log |
```

```
+-----+
```

```
real 0m4.682s
```

```
user 0m0.795s
```

```
sys 0m0.585s
```

Monday, 20 July 2009

You can use the shortcut script for the replication as well.

Here's an example of such call. You will invoke sb, followed by an "r" and the version number. The shortcut script will do everything for you. Create a replication system and then invoke the master.

# more control with replication

```
$ cd $SANDBOX_HOME/rsandbox_5_8_82
$ ./use_all "SELECT @@server_id"
# master
@@server_id
1
# server: 1:
@@server_id
101
# server: 2:
@@server_id
102
```

Monday, 20 July 2009

For multiple sandboxes, you have additional scripts. There are scripts for starting, stopping, and cleaning up all the depending sandboxes at once. And there is a script that lets you run the same SQL statement in all the nodes. In this example, you see the query "SELECT @@server\_id" executed by the master and the slaves in sequence.

# Self contained

- All sandboxes are created inside \$SANDBOX\_HOME
- Use all at once
- Great control



<http://mysqlsandbox.net>

Monday, 20 July 2009

Using MySQL Sandbox servers is safe and easy. Stopping and starting all the servers at once is a simple script call. Since all sandboxes are under the same tree, it's also easy to keep track of storage. There is much more that you can do. Check the MySQL Sandbox site for the reference manual and a detailed cookbook with practical guidance.