

Lab Manual – Relation Database

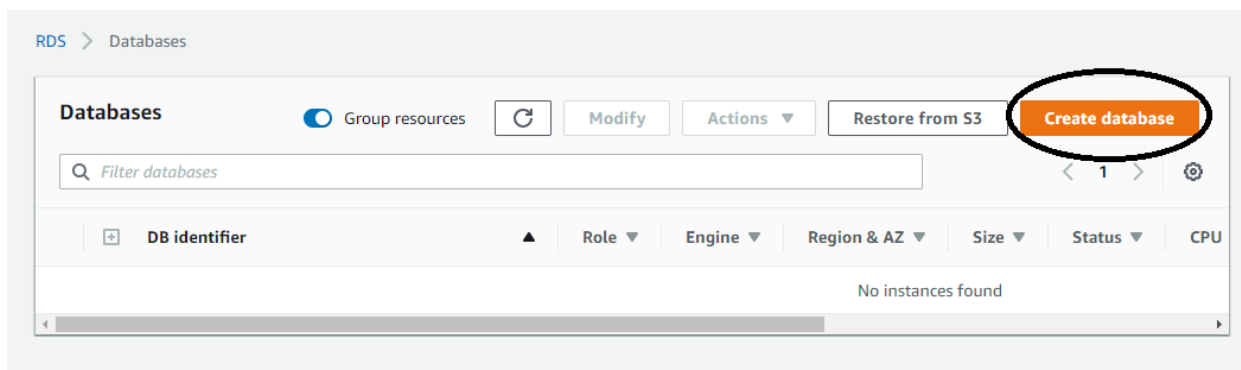
Step

1. Create MYSQL DB Instance.
2. Output
3. Access the MYSQL DB from the public windows machine.
4. Access the MYSQL Db from Linux EC2 in the same VPC

Step1: Create MYSQL DB Instance.

Go to “RDS” service

On the below screen click on “**Create Database**”



RDS > Create database

Create database

Choose a database creation method [Info](#)

☒ **Standard Create**

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

☐ **Easy Create**

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

☐ Amazon Aurora



☒ **MySQL**



☐ MariaDB



☐ PostgreSQL



☐ Oracle

ORACLE®

☐ Microsoft SQL Server



Edition

☒ MySQL Community

Version [Info](#)

MySQL 5.7.22

MySQL 5.6.44

MySQL 5.7.16

MySQL 5.7.17

mysql 5.7.19

mysql 5.7.21

MySQL 5.7.22

MySQL 5.7.23

MySQL 5.7.24

MySQL 5.7.25

ability issues with specific

From the list, select the required version of MYSQL, This is a PAAS solution.

Templates

Choose a sample template to meet your use case.

- ☐ **Production**
Use defaults for high availability and fast, consistent performance.
- ☒ **Dev/Test**
This instance is intended for development use outside of a production environment.
- ☐ **Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

Now , lets assign a username and passwd for the database

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

☐ Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), " (double quote) and @

Confirm password [Info](#)

Note: -- we did not assign an key, which means we DO NOT HAVE ACCESS TO THE UNDERLINE VM hosting the MYSQL database.

DB instance size

DB instance class [Info](#)

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

- ☒ Standard classes (includes m classes)
- ☐ Memory Optimized classes (includes r and x classes)
- ☐ Burstable classes (includes t classes)

db.m5.xlarge

4 vCPUs 16 GiB RAM EBS: 3500 Mbps

☐ Include previous generation classes

Storage

Storage type [Info](#)

General Purpose (SSD)

Allocated storage

20

GiB

(Minimum: 20 GiB, Maximum: 65536 GiB) Higher allocated storage [may improve](#) IOPS performance.

i Provisioning less than 100 GiB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. [external link for more details.](#)

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

☒ Enable storage autoscaling

Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold

1000

GiB

Minimum: 21 GiB, Maximum: 65536 GiB

Availability & durability

Multi-AZ deployment [Info](#)

- ☐ Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
- ☒ Do not create a standby instance

This option is for creating standby instance for High Availability.

Note: -- NOT A READ REPLICA.

Connectivity



Virtual Private Cloud (VPC) [Info](#)

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-0b17f9a14de147cc1) ▼

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

▼ Additional connectivity configuration

Subnet group [Info](#)

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default ▼

We could configure the Subnet group to have selected subnets that we need and the DB would be part of it.

▼ Additional connectivity configuration

Subnet group [Info](#)

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default ▼

Publicly accessible [Info](#)

☒ Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☐ No

RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

This option is to make the DB publicly available.

VPC security group

Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

☐

Choose existing

Choose existing VPC security groups

☒

Create new

Create new VPC security group

New VPC security group name

RDS-MYSQL-SG01

Availability zone [Info](#)

us-east-1a ▼

Database port [Info](#)

TCP/IP port the database will use for application connections.

3306

This would create a new security group for the DB instance and the DB port is also changeable.

Addition configuration that can be configured are as below.

1. Backup configuration
2. DB parameters Group.
3. Encryptions
4. Performance insights Retention period
5. Monitoring
6. Log Exports
7. Maintenance
8. Delete protections

Step2: Outputs.

Successfully created database [database-1](#).

View credential details

RDS > Databases > database-1

database-1

Modify

Actions

Summary

DB identifier database-1	CPU <div>0.00%</div>	Info <div>Available</div>	Class db.m5.xlarge
Role Instance	Current activity <div>0 Sessions</div>	Engine MySQL Community	Region & AZ us-east-1a

Connectivity & security

Endpoint & port

Endpoint
database-1.cnhqgkxvqwd4.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability zone
us-east-1a

VPC
vpc-0b17f9a14de147cc1

Subnet group
default

Subnets
subnet-0f3b774d3d05f9a3b
subnet-0ec726ff4ca782a50
subnet-0b8044d04915fd918
subnet-05804d6144038ec52
subnet-04f51f4d2fe464bc2
subnet-068f945b1fd13595a

Security

VPC security groups

RDS-MYSQL-SG01 (sg-0b7f75645dd5aa8b1) (active)

Public accessibility
Yes

Certificate authority
rds-ca-2015

Certificate authority date
Mar 6th, 2020

The security group is automatically created, with below

Lab Manual – Relational Database

search : sg-0b7f75645dd5aa8b1

Name	Group ID	Group Name	VPC ID	Owner
	sg-0b7f75645dd5aa8b1	RDS-MYSQL-SG01	vpc-0b17f9a14de147cc1	573625096654

Security Group: sg-0b7f75645dd5aa8b1

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source	Description
MySQL/Aurora	TCP	3306	42.109.159.15/32	

This is public ip address of my laptop that is taken automatically

This shows, that the AWS is by default securing the DB to be accessed only from my Laptop, For, it be accessed by the EC2 instance in the same VPC, we would need to add in the inbound rules.

Security Group: sg-0b7f75645dd5aa8b1

Description Inbound Outbound Tags

Edit

Type	Protocol	Port Range	Source	Description
MySQL/Aurora	TCP	3306	42.109.159.15/32	
MySQL/Aurora	TCP	3306	172.31.0.0/16	VPC CIDR

There is no replicated DB instance, as we did not select the same.

Replication (1)

Filter replication

DB instance	Role	Zone	Replication source
database-1	Instance	us-east-1a	-

Even the snapshots are maintained automatically.

RDS > Snapshots > rds:database-1-2019-09-05-12-59

rds:database-1-2019-09-05-12-59

Details

ARN	Snapshot creation time
arn:aws:rds:us-east-1:573625096654:snapshot:rds:database-1-2019-09-05-12-59	Thu Sep 05 18:30:53 GMT+530 2019
Instance/Cluster Name	Instance/Cluster Creation
database-1	Thu Sep 05 18:29:36 GMT+530 2019
Master username	VPC
admin	vpc-0b17f9a14de147cc1
DB snapshot name	Status
rds:database-1-2019-09-05-12-59	available
Snapshot type	Storage type
automated	General Purpose (SSD)

Dashboard
Databases
Query Editor
Performance Insights
Snapshots
Automated backups
Reserved instances
Subnet groups

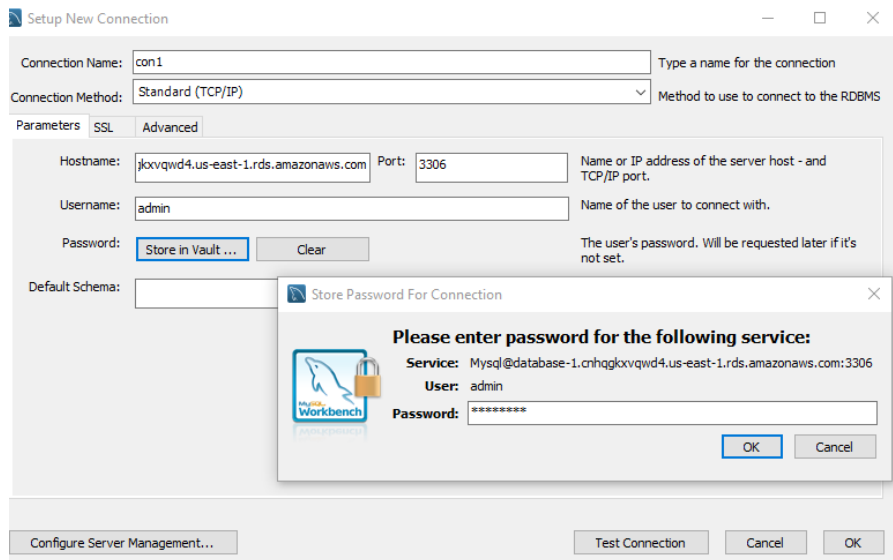
Active backups (1)					
Filter active backups					
DB Name ▲	Earliest restorable time	Latest restorable time	Engine ▼	Encrypted ▼	
<input type="radio"/> database-1	Thu Sep 05 2019 18:30:53 GMT+0530	Thu Sep 05 2019 20:35:00 GMT+0530	mysql	Yes	

If we observe carefully, the automated backup is running every hour and it is showing the earliest and the latest restorable times.

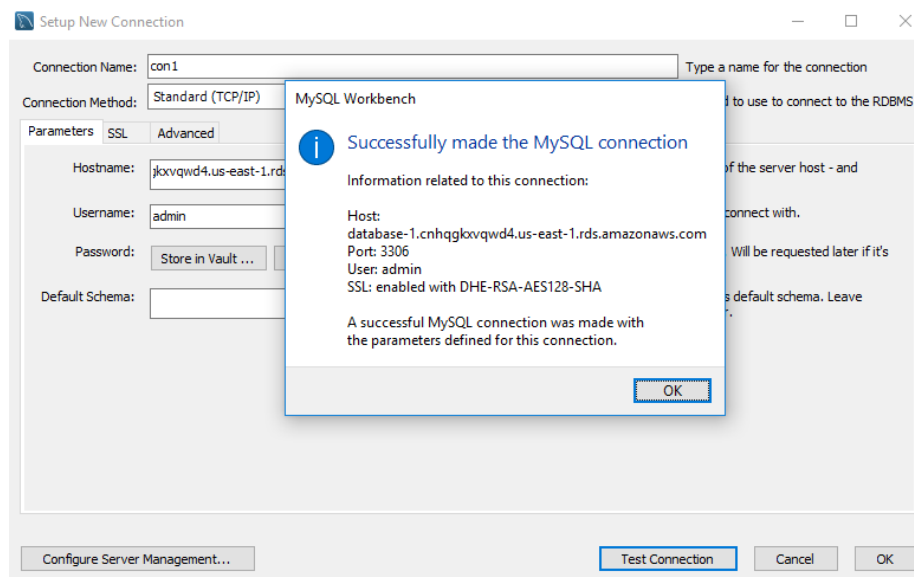
Step3: Access the MYSQL DB from the public.

Here, we are using Windows machine with MYSQL Workbench

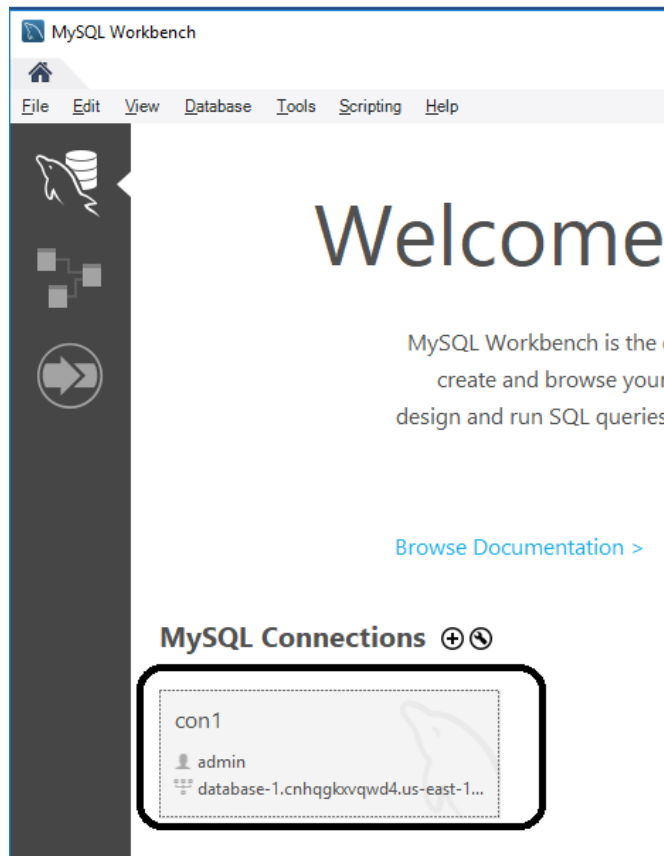
In the hostname: input the Endpoint name of the DB.



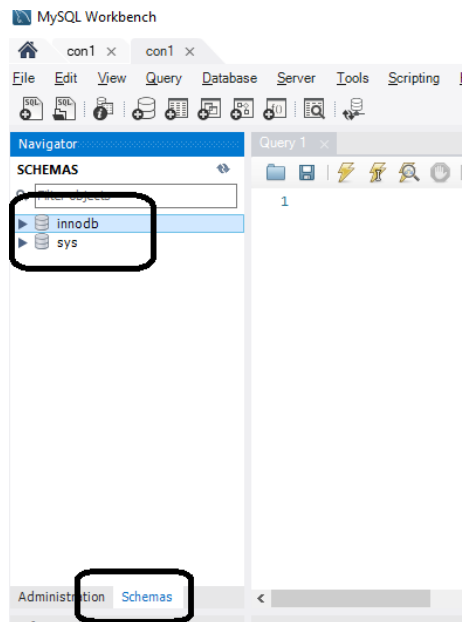
Click on “Test Connection” to check the connectivity



This shows that the DB connection from internet is fine.

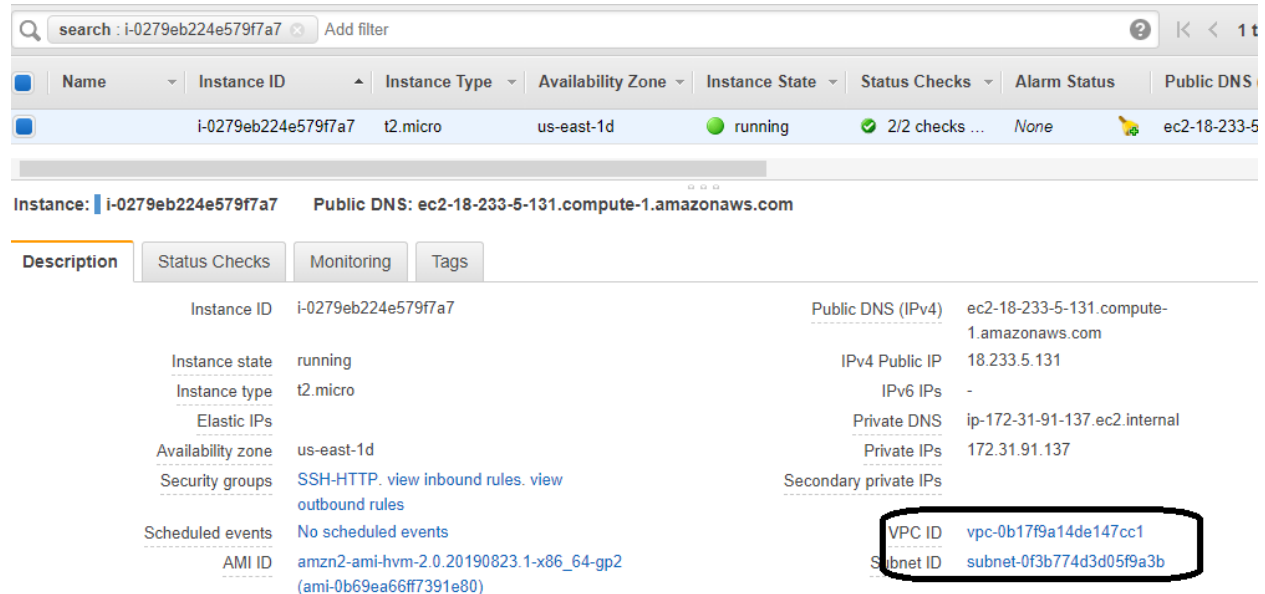


Double click on the icon that is highlighted.



The default DB's are listed here.

Step4: Access the MYSQL Db from Linux EC2 in the same VPC



The screenshot shows the AWS Management Console interface for an EC2 instance. The instance is named 'i-0279eb224e579f7a7', is of type 't2.micro', and is in the 'us-east-1d' availability zone. It is currently in a 'running' state. The public DNS is 'ec2-18-233-5-131.compute-1.amazonaws.com'. The instance is associated with the 'SSH-HTTP' security group. The VPC ID is 'vpc-0b17f9a14de147cc1' and the Subnet ID is 'subnet-0f3b774d3d05f9a3b'.

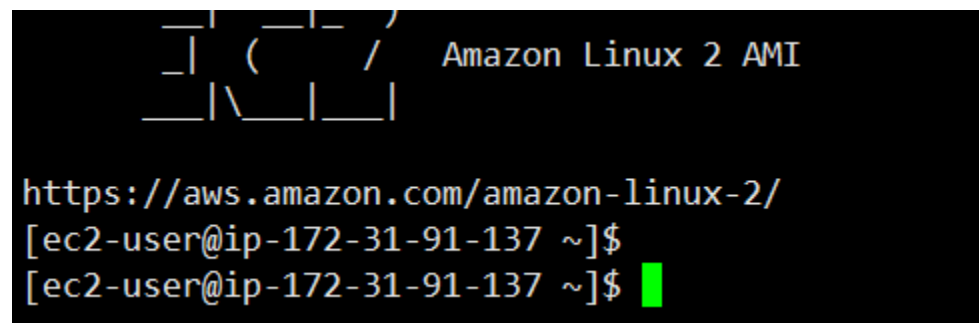
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-0279eb224e579f7a7	t2.micro	us-east-1d	running	2/2 checks ...	None	ec2-18-233-5-131.compute-1.amazonaws.com

Instance: **i-0279eb224e579f7a7** Public DNS: **ec2-18-233-5-131.compute-1.amazonaws.com**

Description | Status Checks | Monitoring | Tags

Property	Value
Instance ID	i-0279eb224e579f7a7
Instance state	running
Instance type	t2.micro
Elastic IPs	
Availability zone	us-east-1d
Security groups	SSH-HTTP, view inbound rules, view outbound rules
Scheduled events	No scheduled events
AMI ID	amzn2-ami-hvm-2.0.20190823.1-x86_64-gp2 (ami-0b69ea66f7391e80)
Public DNS (IPv4)	ec2-18-233-5-131.compute-1.amazonaws.com
IPv4 Public IP	18.233.5.131
IPv6 IPs	-
Private DNS	ip-172-31-91-137.ec2.internal
Private IPs	172.31.91.137
Secondary private IPs	
VPC ID	vpc-0b17f9a14de147cc1
Subnet ID	subnet-0f3b774d3d05f9a3b

Created an EC2 instance in the same VPC as the DB instance.



```
[ec2-user@ip-172-31-91-137 ~]$ ping database-1.cnhqgkxvqwd4.us-east-1.rds.amazonaws.com
PING ec2-52-204-175-214.compute-1.amazonaws.com (172.31.18.127) 56(84) bytes of data:
^C
--- ec2-52-204-175-214.compute-1.amazonaws.com ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1006ms
[ec2-user@ip-172-31-91-137 ~]$
```

Here we are trying to ping the dns name of the DB instance

The ping will not reply as the DB instance block it.

But, what we have to observe is , that the DNS resolves to an private address, where in if we ping from the public network it would resolve to the public ip as below.

```
[c:\~]$ ping database-1.cnhqgkxvqwd4.us-east-1.rds.amazonaws.com
Pinging ec2-52-204-175-214.compute-1.amazonaws.com [64:ff9b::34cc:afd6] with 32 bytes of data:
^C
[c:\~]$ ping -4 database-1.cnhqgkxvqwd4.us-east-1.rds.amazonaws.com
Pinging ec2-52-204-175-214.compute-1.amazonaws.com [52.204.175.214] with 32 bytes of data:
^C
[c:\~]$ █
```

In fact, its resolving to both IPv4 and IPv6 public ip's.

Let's login to the DB instance.

```
[ec2-user@ip-172-31-91-137 ~]$ mysql -h database-1.cnhqgkxvqwd4.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 74
Server version: 5.7.22-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> █
```

```
MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| innodb |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

MySQL [(none)]> █
```

Step5: Create a new DB

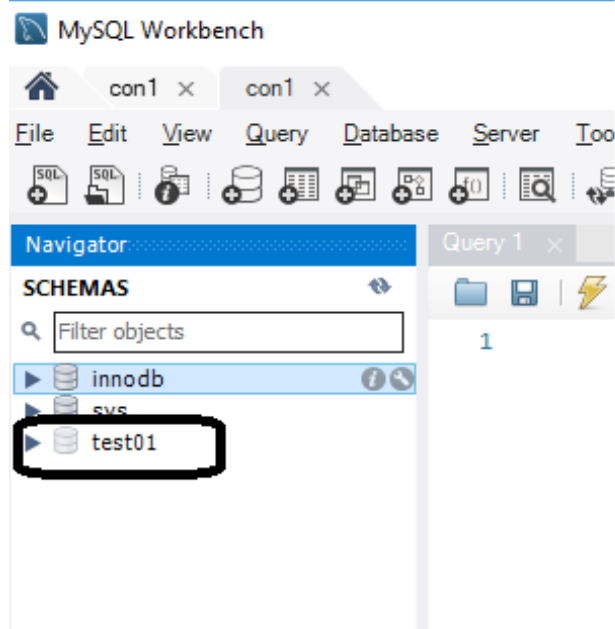
Create database <<database name>>

```
MySQL [(none)]> create database test01;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| innodb      |
| mysql      |
| performance_schema |
| sys        |
| test01     |
+-----+
6 rows in set (0.00 sec)

MySQL [(none)]> █
```

Lets check in the workbench



This shows, that the same DB can be easily accessed from private and public network.

