Database Options:  
**Database Name:** (optional) The name of your Postgres database to be created.If you do not provide a default will be created automatically.  
**Database Port:** The port you want to use to access the database. PostgreSQL default is port 5432.  
**DB Parameter Group:** A configuration applied to the database. If you haven’t created one then choose the default.  
**Option Group:** More options of the database. Use the default value unless you have created your own option group.  
**Copy Tags To Snapshots:** check if you want the tags metadata on DB instances copied to corresponding instance snapshots.  
**Enable Encryption:** Specify if you want the database and snapshots to be encrypted.

Backup:  
**Backup Retention Period:** Set the number of days you want automatic backups of your database to be retained. Always keep on mind on the cost that this may incurr.  
**Backup Window:** Select the time of day you’d like you backup to happen.

Maintenance:  
**Auto Minor Version Upgrade:** Select to enable your DB instance to receive minor DB engine version upgrades automatically when they become available.  
**Maintenance Window:** Select a 30 minute time window that you prefer updates and other maintenance to happen. If time period doesn’t matter, select **No Preference**.

10. When you’re finished with the setup, click **Launch DB Instance**. On the final page of the wizard, click **Close**.

11. On the Amazon RDS console, at Instances tab you should see the new DB instance appears in the list. When its status is **Available** then it is ready for use.

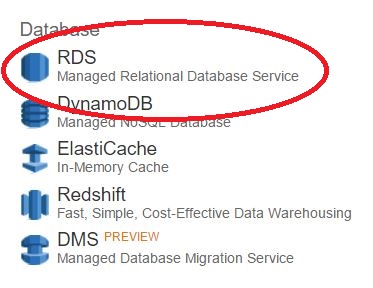
**Connect to an AWS RDS instance inside a VPC using MySQL Workbench**

**In this step by step how-to we are going to setup a private RDS instance in the default VPC and a bastion host to open an access to the RDS instance from the Internet without the need of a VPN connection.**

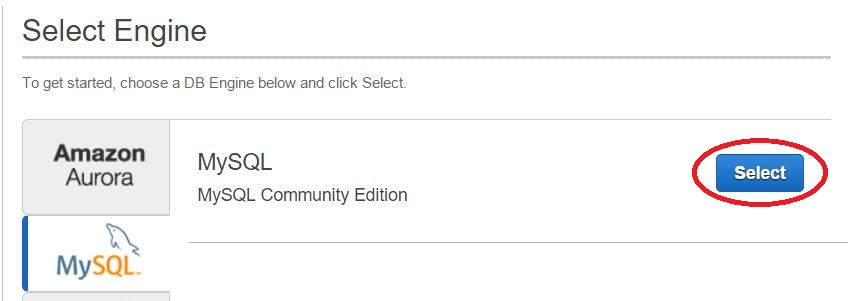
VPCs are very powerful in securing your cloud infrastructure but they are also making harder to connect to an instance from outside the VPC to import or backup data or for maintenance. We are going to see how to access an AWS RDS instance remotely.

In this article we will:  
- Use the default VPC.  
- Create a new private (not publicly accessible) RDS instance in the default VPC.  
- Create a new Linux  bastion host in the default VPC.

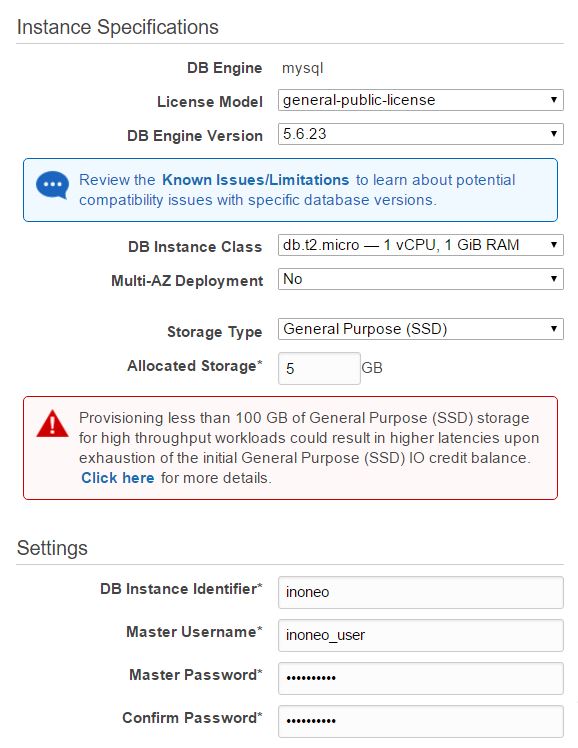
**Step 1: Create an RDS instance inside a VPC**

**[](http://static.inoneo.com/aws/connect_vpc_rds/rds.jpg)**

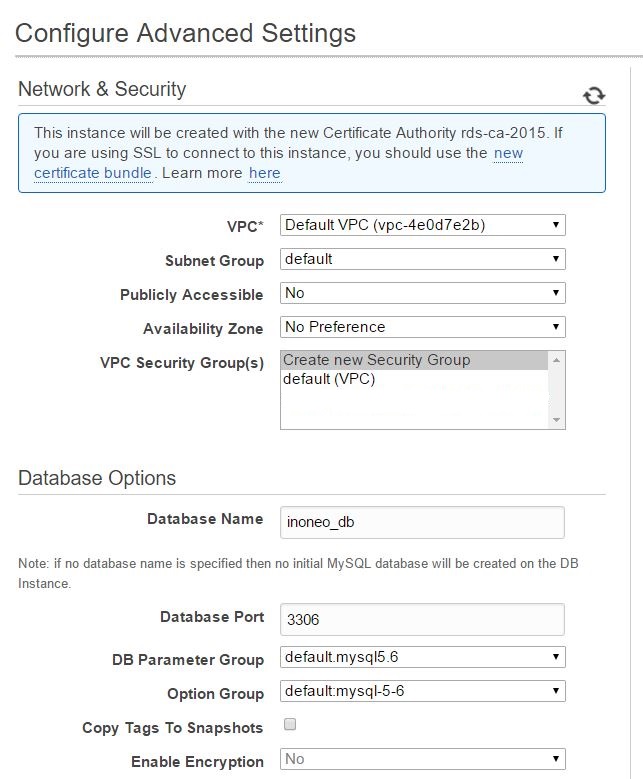
Select the DB engine. For this example we will use **MySQL**. If you use a different engine, steps will be the same, only the port will differ.

**[](http://static.inoneo.com/aws/connect_vpc_rds/mysql.jpg)**

Instance specifications

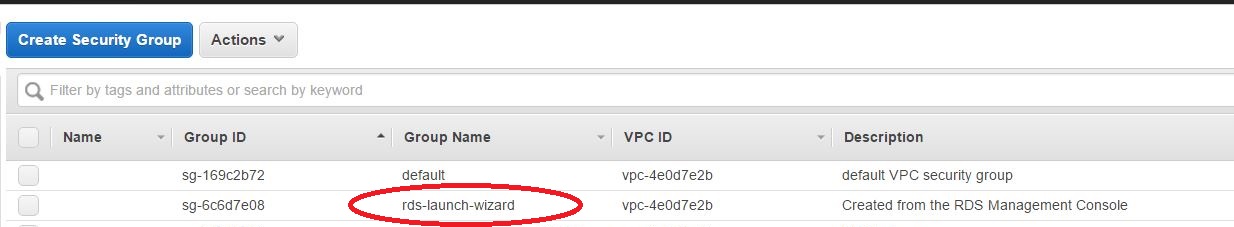
**[](http://static.inoneo.com/aws/connect_vpc_rds/db_setup.jpg)**

DB specifications

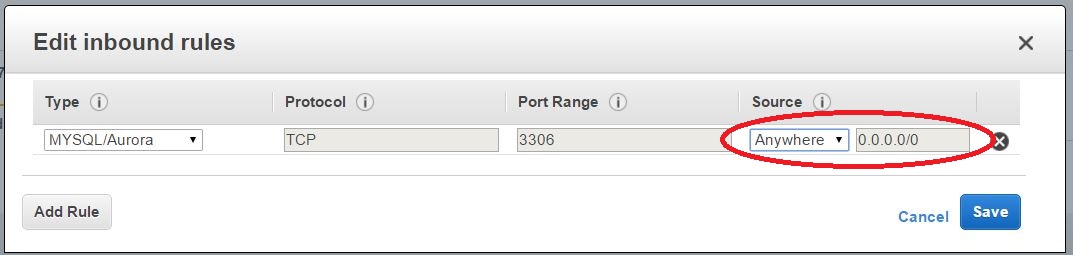
**[](http://static.inoneo.com/aws/connect_vpc_rds/db_setup_2.jpg)**

Be sure to select **No** to public access.  
Launch the DB creation and navigate to the EC2 console. On the left pane select **security groups**:

You should see a newly created security rule called **‘rds-launch-wizard’**, select it.

**[](http://static.inoneo.com/aws/connect_vpc_rds/rds-launch-wizard-sec-group.jpg)**

In the **inbound tab**, select the source for MySQL TCP to **anywhere** (it will be accessible only inside the VPC).

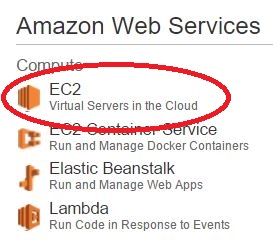
**[](http://static.inoneo.com/aws/connect_vpc_rds/sec-group-3600-rule.jpg)**

**Note**: In production you should restrict the source by IP address (allow only instances that need to access RDS for added security).

**Step 2: Set up a bastion host**

Because the Amazon RDS instance is not publicly accessible, you won’t be able to access it from outside (from Internet).  
You have different options to get access to the RDS instance like creating a **VPN** or adding a **bastion host**. We’ll choose the latter one because it’s quick to setup and we don’t have to use a VPN client.

Go back to the AWS console and select EC2 under compute:

**[](http://static.inoneo.com/aws/connect_vpc_rds/ec2.jpg)**

Then select Launch instance:

**[](http://static.inoneo.com/aws/connect_vpc_rds/launch-ec2-instance.jpg)**

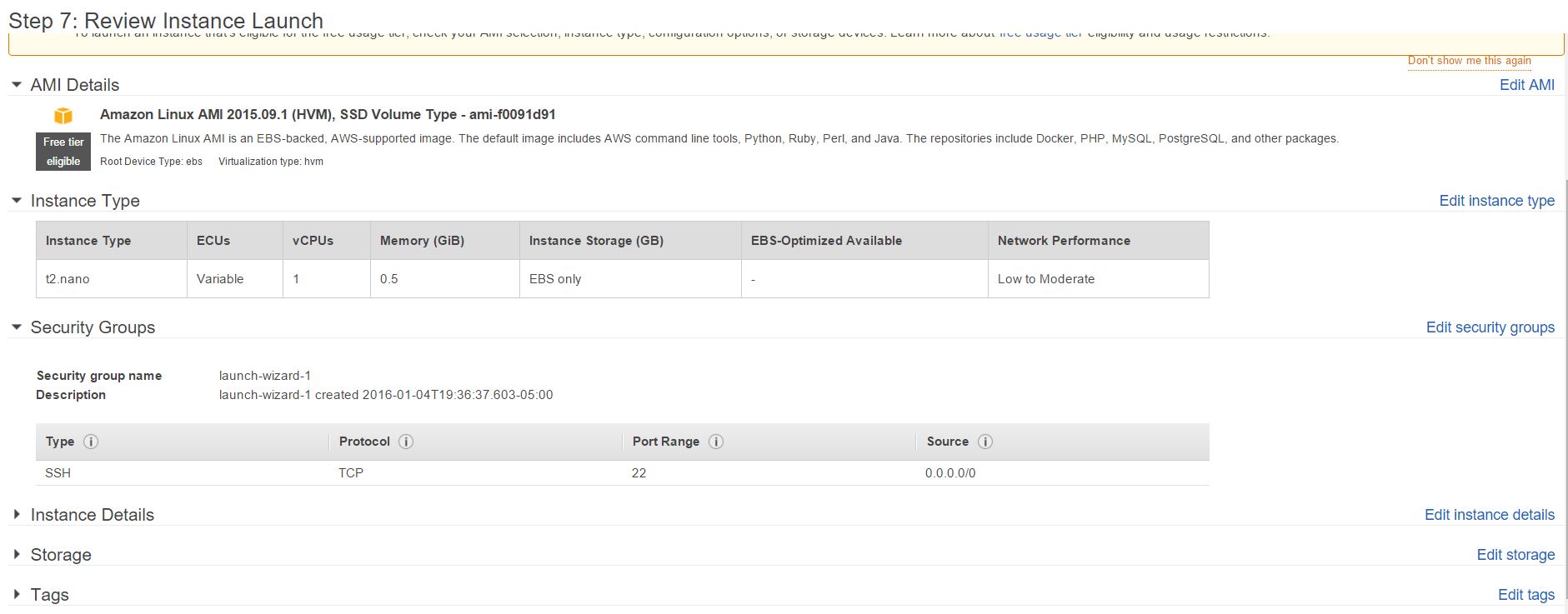
For the bastion host we’ll choose the Amazon Linux AMI:

**[](http://static.inoneo.com/aws/connect_vpc_rds/amazon-ami.jpg)**

Choose the cheapest option, like a **t2.nano**. This instance will only be used to ssh into the VPC.

Select the default VPC.

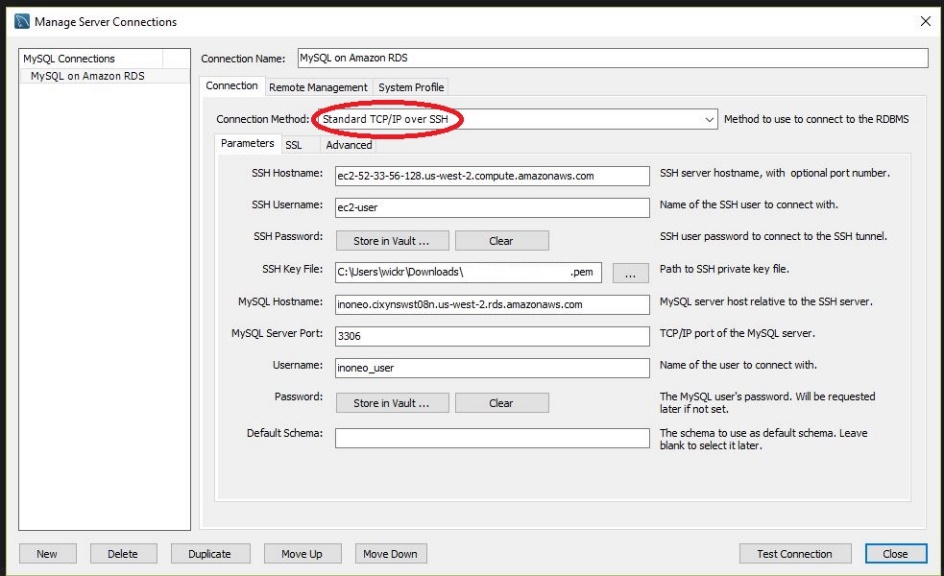
For this demo we will open **22** to the world but in reality you should restrict the ssh access to your public IP address only(source of the security group for TCP 22).

**[](http://static.inoneo.com/aws/connect_vpc_rds/review-and-launch.jpg)**

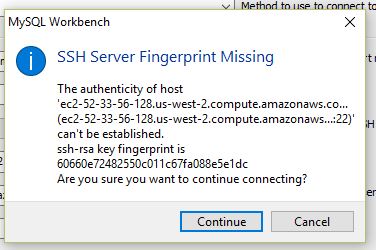
**Step 3: Configure a new connection in MySQL Workbench**

Create a new connection.

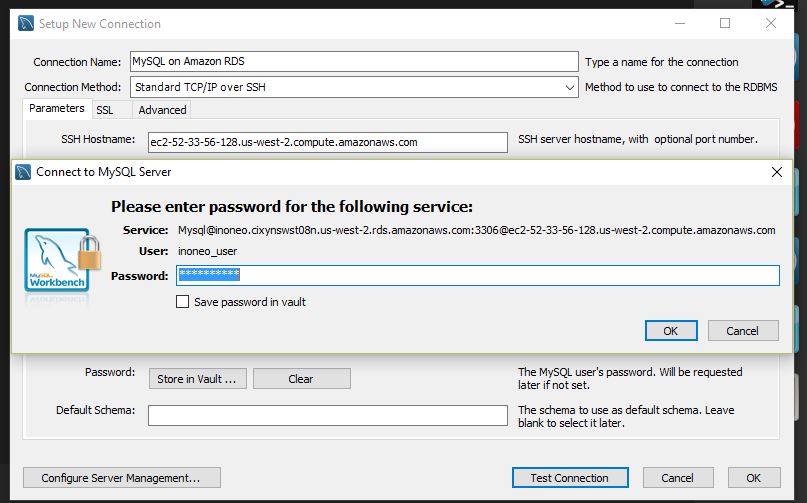
Select **Standard TCP/IP over SSH**.  
**SSH Hostname** : Enter your bastion host public DNS  
**SSH username**: ec2-user  
**SSH Key** : Select the bastion private key.  
**MySQL Hostname**: Enter your RDS DNS name  
**MySQL port** : 3306  
**Username**: Your DB username created during the RDS launch wizard.

**[](http://static.inoneo.com/aws/connect_vpc_rds/good-connection-details.jpg)**

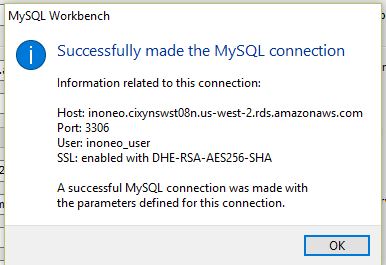
Click on Test connection:

**[](http://static.inoneo.com/aws/connect_vpc_rds/ssh-fingerprint.jpg)**

If it’s your first SSH connection to the bastion host instance from MySQL Workbench, it will ask to add a **SSH Server fingerprint**, click on Continue.

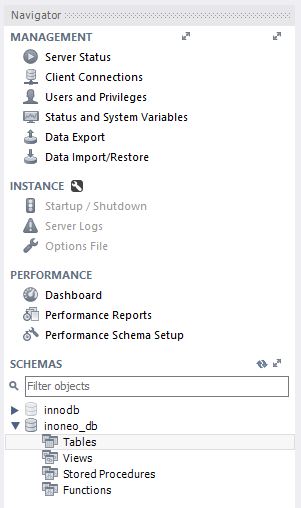
**[](http://static.inoneo.com/aws/connect_vpc_rds/input_password.jpg)**

Enter your Amazon RDS DB password.

**[](http://static.inoneo.com/aws/connect_vpc_rds/success.jpg)**

You should have a **success message** telling that MySQL Workbench is able to connect to the Amazon RDS instance.

Finally you can save and open a connection to the MySQL database and list all databases, tables. Etc..

**[](http://static.inoneo.com/aws/connect_vpc_rds/db-pane.jpg)**

**Conclusion**

In this example we used the **default VPC** to simplify the steps. In a real environment your RDS instance will reside in a private subnet and the bastion host should be in the public subnet. The only difference is to configure an ACL to allow access from the bastion host to the private subnet on port 22 (for ssh).