

Introduction

This exercise is aimed at introducing you to a couple of important topics in the Data space, if you are not already familiar with them. **Cloud computing** and **Relational Databases**. There are a variety of options available in the market for these, provided by different vendors. For this class particularly, we are going to use Amazon's Cloud offering "AWS" (Amazon Web Services) and MySQL database that runs in Amazon's cloud (Amazon cloud, in simple terms, is nothing but a really big Data Center, with a ton of computers, that belongs to Amazon). Amazon calls their managed service for Relational Databases as RDS (Relational Database Service). Amazon offers a bunch of other services in the cloud as part of AWS.

AWS Account creation

1. Create an AWS Personal account at <http://aws.amazon.com/>
2. Provide credit card info. (**Note:** AWS uses your payment information to verify your identity and only for usage in excess of the [AWS Free Tier Limits](#). AWS will not charge you for usage below the AWS Free Tier Limits.) *For this assignment make sure to **ONLY** use free tier options when launching any new AWS resources.
3. Choose "Basic Plan" (FREE)

Launching a MySQL Database Instance in your AWS account

1. Sign-in to the management console from <https://aws.amazon.com/> (My Account -> AWS Management Console)
2. Find services -> RDS (Managed Relational Database Service)
3. Create database
4. **Creation method=Standard create, Engine=MySQL, Template=Free tier,**

Free Tier

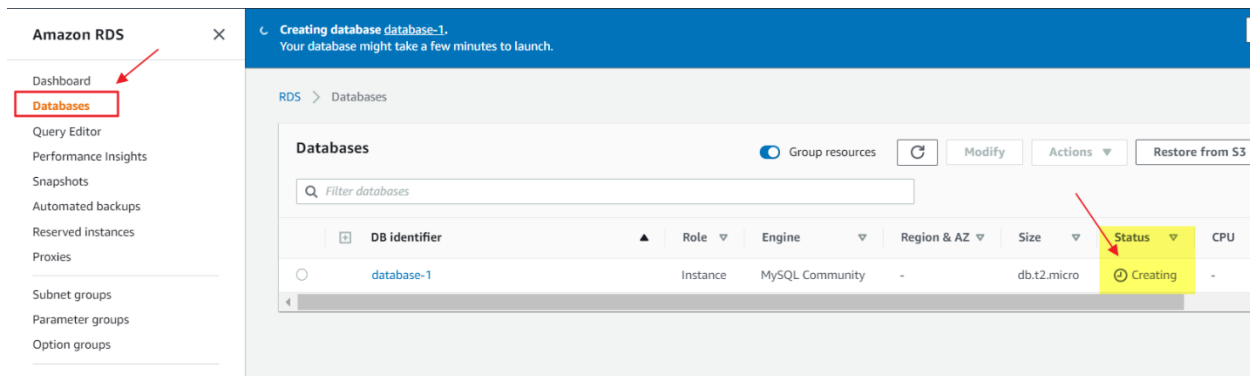
The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier.](#)

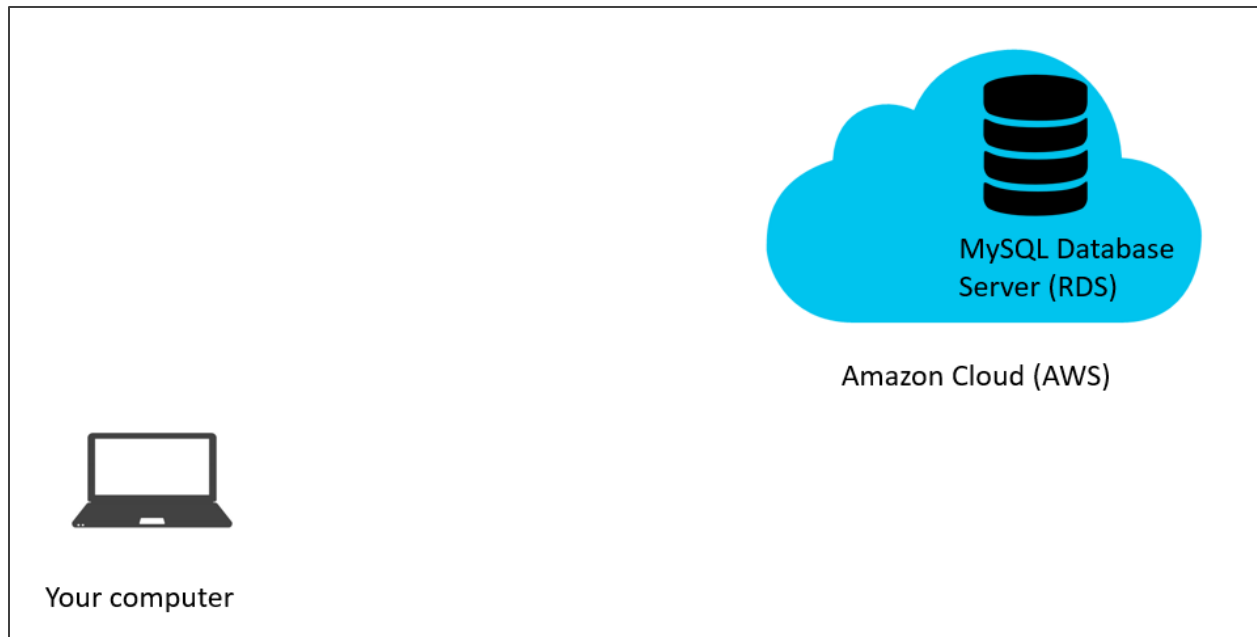
When you free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

5. Fill out the settings for DB Instance identified and master password
6. Leave everything else as the default settings proposed by AWS. Changing any of those settings may impact the free tier usage and cost you money.
7. Click create database.



So, you have just launched a MySQL Database Server in the Amazon Cloud. Note that Amazon offers a lot of other services than just Relational Databases, such as Web Servers, Application servers, Hadoop Clusters, Storage services etc. just to name a few. With simple clicks of buttons on the Amazon Website (console) you can launch any of these services from your computer, these services run on Amazon's hardware, Amazon takes care of maintaining that hardware and also the software, you just pay for the usage.

Arch Talents LLC



But how do you connect to this Database if you want to do anything on it?

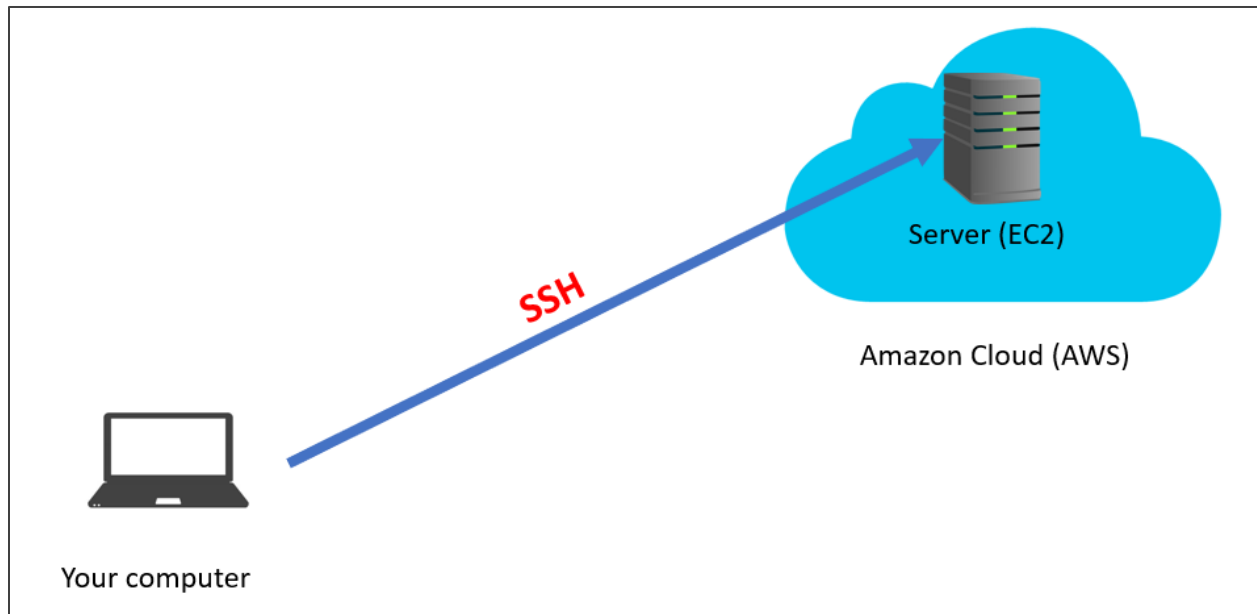
Well, how do you connect to anything that runs in AWS?

If you are running a server (like a web server or an app server) on AWS, then you can remotely connect to it using SSH (please read more about this online if you are interested).

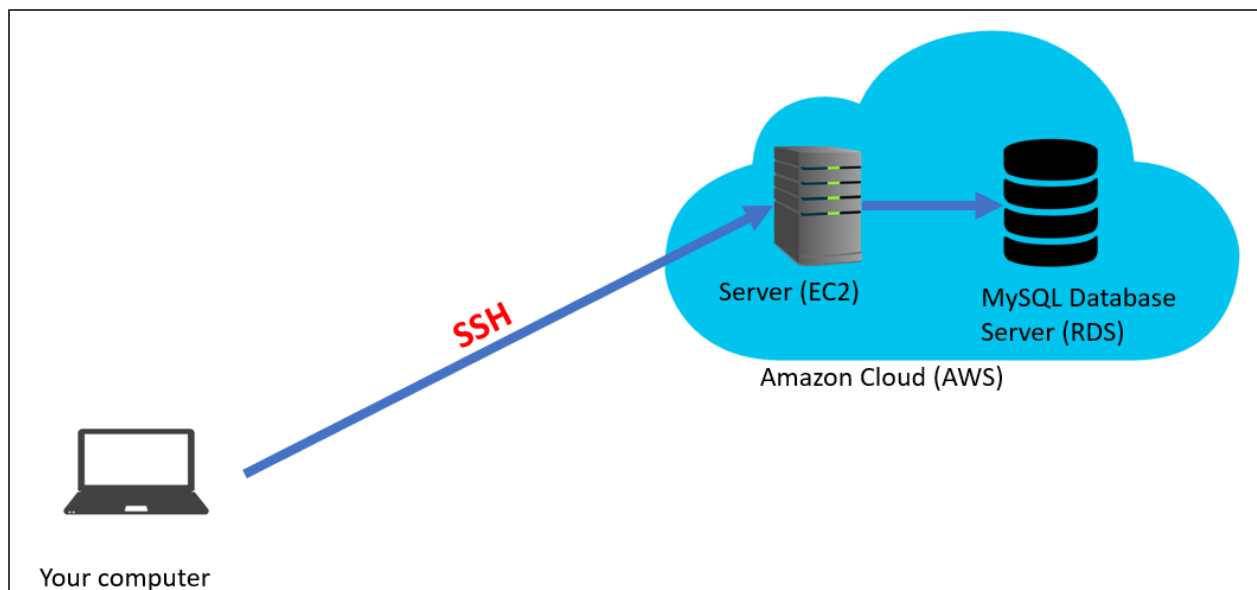
In simple words, with SSH you basically type commands on the command line on your computer, but they execute on the remote AWS server.

By the way, the Amazon service for servers is called EC2.

Arch Talents LLC



So, can you SSH from your computer to RDS then? Well, AWS doesn't allow direct SSH to RDS instances. You will have to first SSH to an EC2 instance and then “jump” from that EC2 to RDS via a feature in SSH called local port forwarding (do not worry too much about these concepts, you only need to know the correct SSH commands for this and need not understand the underlying concepts).



Launching an EC2 instance

1. Sign-in to the management console from <https://aws.amazon.com/> (My Account -> AWS Management Console)
2. Find services -> EC2
3. Instances (on the left menu) -> Launch Instance

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI) Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

Quick Start 1 to 18 of 18 AMIs

My AMIs

AWS Marketplace

Community AMIs

☒ **Free tier only**

Amazon Linux **Free tier eligible**

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e38b48473ea57778 (64-bit x86) / ami-0fb3bb3e1ae2da0be (64-bit Arm) Select

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Amazon Linux **Free tier eligible**

Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0998bf58313ab53da Select

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

64-bit (x86)

Pick the latest Amazon Linux instance (in the above example it is Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0e38b48473ea57778 (64-bit x86) / ami-0fb3bb3e1ae2da0be (64-bit Arm)).

4. Make sure to choose the Free tier option and keep clicking on the "Next" step until the end. You don't have to change any settings, just keep hitting Next.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

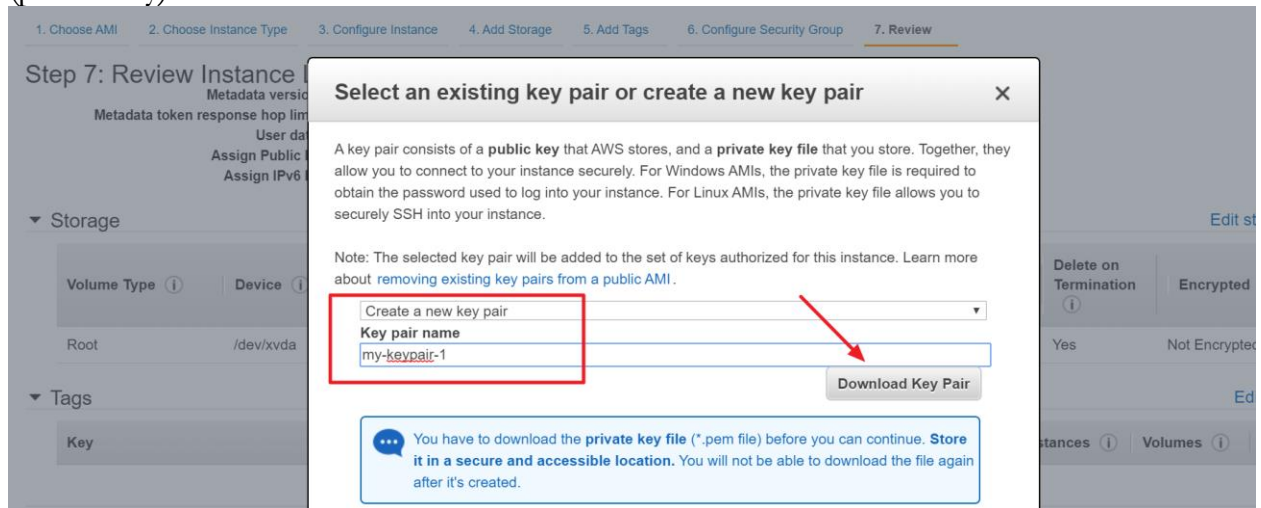
Filter by: **All instance types** **Current generation** [Show/Hide Columns](#)

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

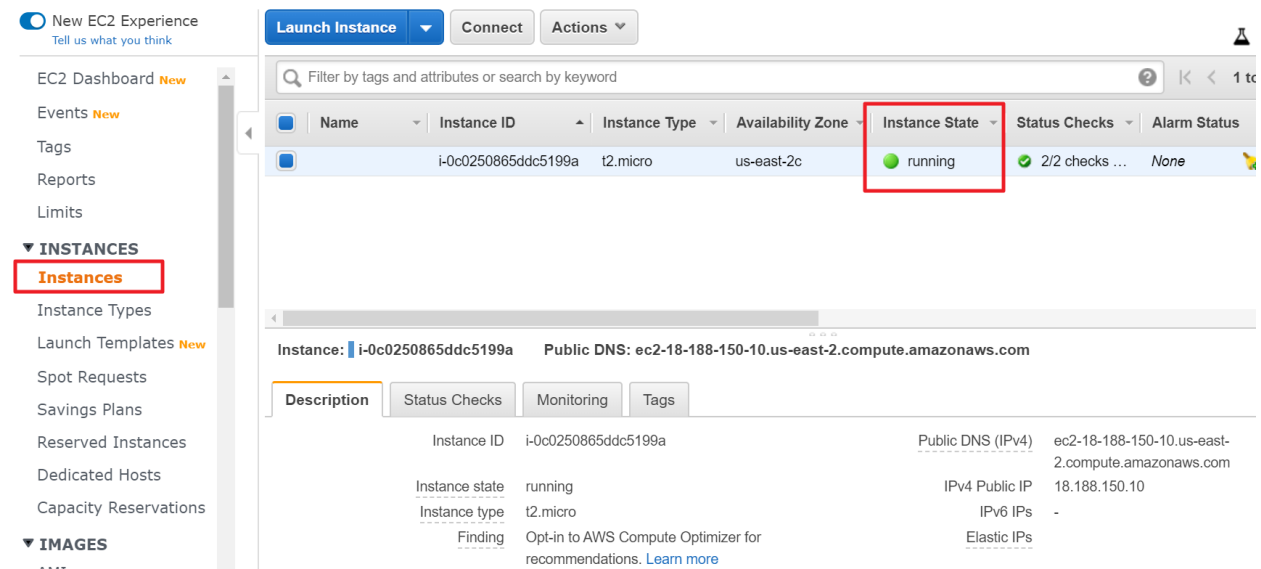
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Detail

- Once you reach the last step and click on “Launch” you will be asked for a “Key Pair”. Choose the option to create a new one. You will need to download the “.pem file” for your key pair and store it securely. The .pem file is the private key. Remember we said you can SSH into an EC2 server from within your computer? We need the .pem file (private key) in order to SSH into the EC2 instance.



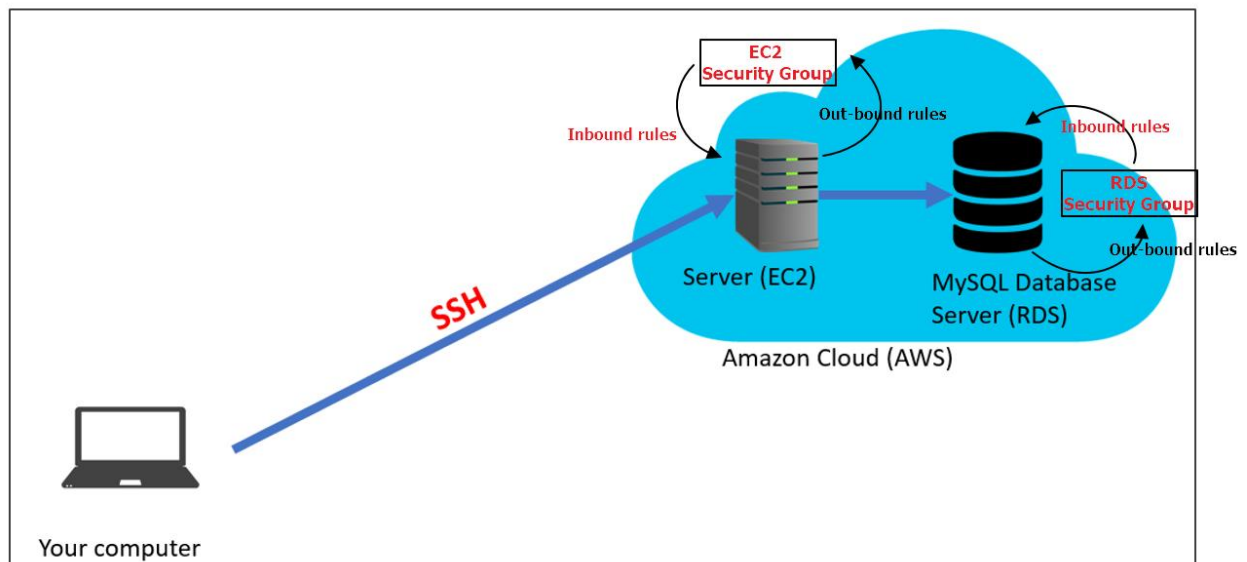
- Once it is launched you can see it from the Instances menu on the left.



Connecting to EC2 instance via SSH

Before we learn how to connect to an EC2 instance via SSH, let's think about security for a moment. Can anyone in the world with a computer connect to your server by default? If the answer is yes then that is a big security hole, we do NOT want that. Instead we want the ability to restrict what computers can connect to your server. For this purpose, AWS has a concept of **Security Groups**. Without going too much into detail, just know that a Security group is a set of rules that define what computers (IP addresses) can connect to your instance. These are “in-bound” rules. Similarly, security groups also have “outbound” rules. The outbound traffic is by default set to “un-restricted” (represented by IP address 0.0.0.0/0). This is because you want to access the internet from within your server.

Both EC2 and RDS instances will have Security Groups to filter traffic coming into and out of them.



For this exercise, for simplicity sake, we are going to allow everything in-bound and out-bound for both EC2 and RDS.

When you create the EC2 and RDS instances the corresponding security groups will automatically get created. The good thing is the EC2 security group

will already be created with the correct settings we need, so there is no need to touch it. For RDS security group though, we will need to make a small change which we will see in the later section about connecting to RDS.

1. Open command prompt on your computer.
2. For windows, type ssh-keygen, keep clicking enter like in the screenshot below. For Mac please lookup on Google how to generate SSH public key.

```
C:\Users\sande>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\sande/.ssh/id_rsa):
Created directory 'C:\Users\sande/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\sande/.ssh/id_rsa.
Your public key has been saved in C:\Users\sande/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Erp0ChX2n8iy3G0b1+1fMe9N78/tH/6fDsQw570CtU0 sande@LAPTOP-KKBJ3D60
The key's randomart image is:
+---[RSA 2048]---+
|  o                |
| . o               |
| . o   o .         |
| . o + . *         |
| . + = S . E o     |
| + B o o . = . o + |
| = . + . . o . + . + |
| . + . . o + B     |
| .                  | oB^
+----[SHA256]-----+
C:\Users\sande>
```

3. In Windows 10 typically the SSH home folder will be C:\Users\<username>\.ssh (it's the same path where your public key gets generated when you run ssh-keygen)
4. Paste the .pem file that you downloaded during EC2 launch (step 5 in the above section) in that location.
5. You can get the correct SSH command to connect to the EC2 from the AWS console (EC2 -> Instances -> Select the instance you need to connect to -> Click on the Connect button)

Arch Talents LLC



A screenshot of the AWS Management Console. On the left is a navigation menu with options like 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', and 'INSTANCES'. The 'INSTANCES' section is expanded, showing 'Instances', 'Instance Types', and 'Launch Templates'. The main area shows a table of EC2 instances. The first instance is selected, and the 'Connect' button in the top toolbar is highlighted with a red box and a red arrow pointing to it. Below the table, the instance details for 'i-0c0250865ddc5199a' are visible, including its Public DNS: 'ec2-18-188-150-10.us-east-2.compute.amazonaws.com'.

A screenshot of the 'Connect to your instance' dialog box in the AWS console. The dialog shows three connection methods: 'A standalone SSH client' (selected), 'Session Manager', and 'EC2 Instance Connect (browser-based SSH connection)'. It provides instructions on how to access the instance using an SSH client, including steps to open an SSH client, locate the private key file, and use the 'chmod' command. The example command is highlighted: `ssh -i "mykeypair-1.pem" ec2-user@ec2-18-188-150-10.us-east-2.compute.amazonaws.com`. The dialog also includes a note about the default username and a link to connection documentation.

6. Copy that command and paste it into the command line on your computer. Note that you will need to specify the correct path for the part that says "mykeypair-1.pem"
In my case the SSH command is:

```
ssh -i "C:\Users\sande\.ssh\mykeypair-1.pem" ec2-user@ec2-18-188-150-10.us-east-2.compute.amazonaws.com
```

7. As you can see in the screenshot below, the SSH connection to the remote server was successful. (To disconnect use the exit command)



```
C:\Users\sande>ssh -i "C:\Users\sande\.ssh\mykeypair-1.pem" ec2-user@ec2-18-188-150-10.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-188-150-10.us-east-2.compute.amazonaws.com (18.188.150.10)' can't be established.
ECDSA key fingerprint is SHA256:ibbMBRpWlH2lpAd1CAz8aPLx6uqia2Ta9hbZkhi/8gg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-18-188-150-10.us-east-2.compute.amazonaws.com,18.188.150.10' (ECDSA) to the list of known hosts.
Last login: Mon Mar 2 04:54:17 2020 from 47-40-29-61.dhcp.stls.mo.charter.com

    _   _          _ 
   / \   \       / \
  /___\   \_____/___\
         |_____|_|_|

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 24 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-40-56 ~]$ exit
logout
Connection to ec2-18-188-150-10.us-east-2.compute.amazonaws.com closed.

C:\Users\sande>
```

Note that in the screenshot above I simply connected to the remote EC2 server and immediately exit-ed. But in general, you would connect to the remote server and run commands, like for example, installing a software on the server or copying files to the server etc.

Connecting to the Database in AWS from your computer (via EC2) to interact with it using SQL

1. Unlike connecting to an EC2 server where you can do so using the command line interface, you will need a tool/software to connect to a database. (Although it is technically possible to interact with a database from the command line it is recommended to use a tool that gives you a graphical/visual view of the stuff in the database).
2. There are multiple client tools available to connect to a MySQL Database. For ease of use we recommend **MySQL Workbench**. You can download it from <https://dev.mysql.com/downloads/workbench/>
I tested “mysql-workbench-community-8.0.12-winx64” on my Windows 10 computer and it works.
Note: If downloading this requires an Oracle account, please create one. It's free.
3. Change the RDS's security group inbound rule to allow traffic from everywhere. (Technically we just need to allow connectivity from our EC2 instance but for simplicity sake let's just allow everything). See below.

Arch Talents LLC



Amazon RDS ×

RDS > Databases > database-1

database-1

Modify Actions

Summary

DB Identifier database-1	CPU 1.50%	Info Available	Class db.t2.micro
Role Instance	Current activity 0 Connections	Engine MySQL Community	Region & AZ us-east-2a

Connectivity & security Monitoring Logs & events Configuration Maintenance & backups Tags

Connectivity & security

Endpoint & port Networking Security

Endpoint
database-1.c3i2t-2.rds.amazonaws.com

Availability zone
us-east-2a

VPC security groups
[test_group1 \(sg-0fcb3a8d529dc87b6\)](#) (active)

Create Security Group Actions

search : sg-0fcb3a8d529dc87b6 Add filter

Name	Group ID	Group Name	VPC ID
	sg-0fcb3a8d529dc87b6	test_group1	vpc-522ce239

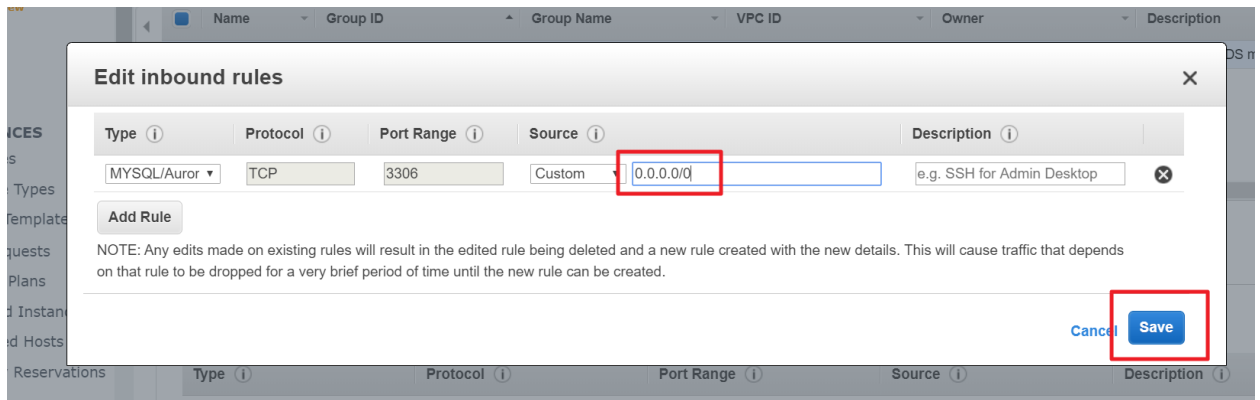
Security Group: sg-0fcb3a8d529dc87b6

Description Inbound Outbound Tags

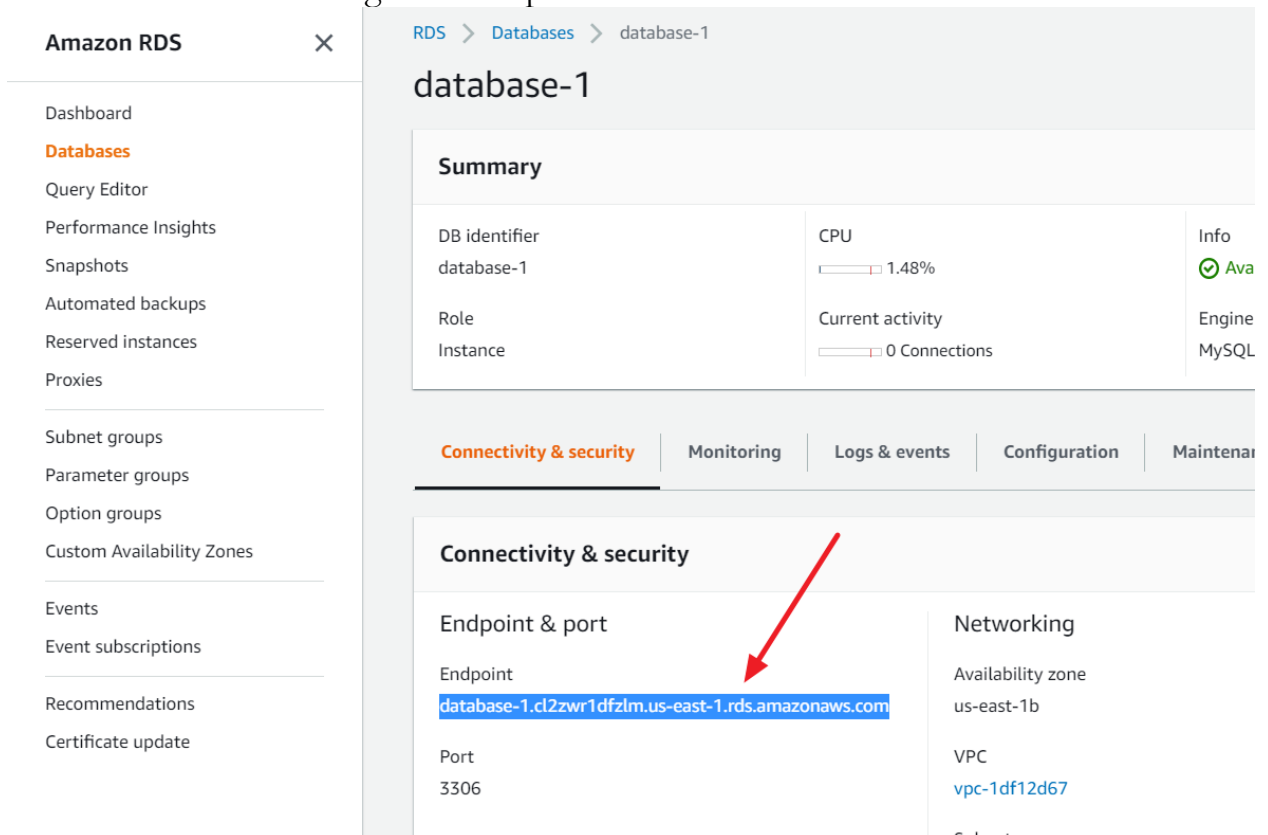
Edit

Type	Protocol	Port Range	Source
MySQL/Aurora	TCP	3306	47.40.1

Arch Talents, LLC
4220 Duncan Avenue, Suite 201, St. Louis, MO 63110,
Tel: 314-884-0477, Fax: 314-754-9474, www.archtalents.com



- Now you need to do the SSH port forwarding. For that you first need to get the RDS “endpoint”.
See screenshot below to get the endpoint:



The syntax for the SSH command is as follows (replace the variables

Arch Talents, LLC
4220 Duncan Avenue, Suite 201, St. Louis, MO 63110,
Tel: 314-884-0477, Fax: 314-754-9474, www.archtalents.com

Arch Talents LLC



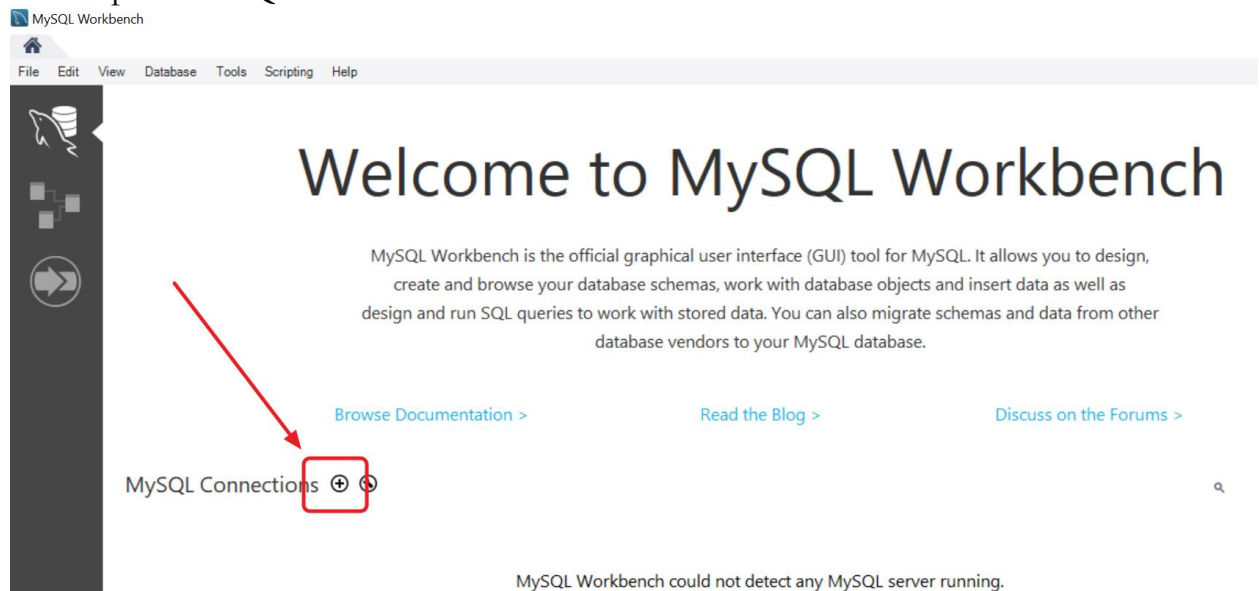
highlighted in different colors below with the actual values)

```
ssh -N -L localPort:rdsHost:remotePort user@remoteEC2Host -i  
~/path/to/key
```

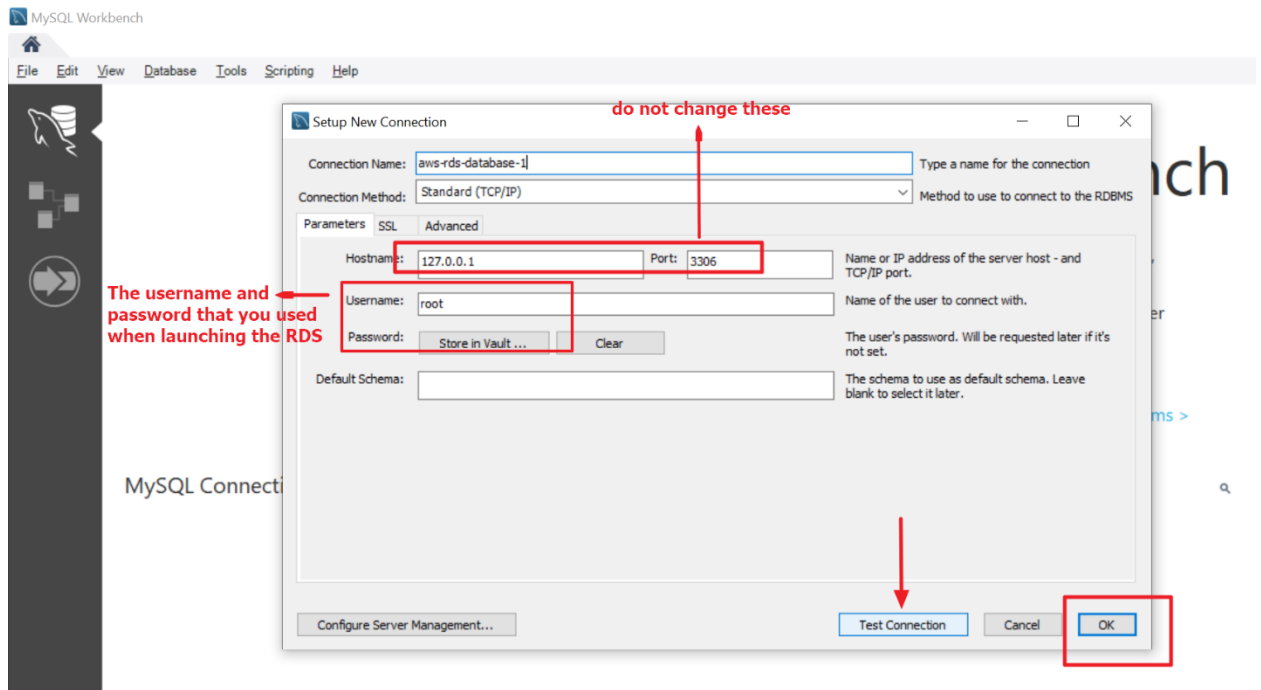
Below is the SSH command I used. Change it accordingly for your use. Just type it in the command prompt and hit enter.

```
ssh -N -L 3306:database-1.cl2zwr1dfzlm.us-east-1.rds.amazonaws.com:3306  
ec2-user@ec2-18-188-150-10.us-east-2.compute.amazonaws.com -i  
"C:\Users\sande\.ssh\mykeypair-1.pem"
```

5. Now open the SQL Workbench



Arch Talents LLC



Arch Talents, LLC
4220 Duncan Avenue, Suite 201, St. Louis, MO 63110,
Tel: 314-884-0477, Fax: 314-754-9474, www.archtalents.com

Practice SQL

Go through the tutorial in the following link and practice SQL statements in your Database.

<https://www.w3schools.com/sql/default.asp>

Here is an example:

Step 1: Create a Database schema:

```
CREATE DATABASE testDB;
```

Step 2: Create a Database Table in the schema you created above:

```
CREATE TABLE testDB.Persons (  
  PersonID int,  
  LastName varchar(255),  
  FirstName varchar(255),  
  Address varchar(255),  
  City varchar(255)  
);
```

Step 3: Inserts records into the table:

```
INSERT INTO testDB.Persons(PersonID, LastName, FirstName,  
Address, City) VALUES (1, 'Ericson', 'Tom', '4006 Some Street', 'Saint Louis');
```

Insert more records using INSERT statements like above

Step 4: Select records from the table:

```
SELECT * FROM testDB.Persons;
```

Note:

If you stuck and need assistance on any of the steps, don't hesitate to reach out to our Data Architect Sai at 734-927-2427, Sai@archtalents.com