**STEP 1 :- Backup Your Wesbite Files**

The very first step of project such as this is to back up every aspect of your website. This step is good practice before any major change but it is also a requirement of migrating your WordPress installation Using an FTP program (such as Transmit), connect to your web host and copy all files under your website’s directory to a folder on your local computer this includes the .htaccess file that is set to be hidden. Consult your FTP program’s help file to have it display hidden files if you are unable to see this file.

**STEP 2 :- Export The WordPress Database**

Exporting your database is a simple process that only requires a few steps to complete. Login to the [cPanel](http://cpanel.net/" \o "cPanel" \t "_blank) account of your web server and open the [phpMyAdmin](https://www.phpmyadmin.net/" \o "phpMyAdmin" \t "_blank) application. Select the database that contains your WordPress installation from the list on the left hand sidebar and once selected click on the Export tab on the navigation menu. The default settings of a Quick export and the SQL format for the export are sufficient for what we need. Click the Go button and the database export process will begin and a file will be downloaded to your local computer.Once the database export and the FTP transfer of your files have both completed, you are ready to move onto the next stage.

**Step 3: Create The WordPress Database On Your New Host Server**

Before we can begin the migration to the new web host, we need to create an environment for a WordPress installation. To do this you must create a database that you can import your SQL data into. Login to your new web host with the user credentials they have supplied you and connect to the cPanel software. For our guide we will be using the MySQL Databases application. If your web host doesn’t have that application running then you will should contact their support team to discover their method of creating new databases.

The steps to create a database are quite simple:

* Open MySQL Database and create a new database with an appropriate name for your website.
* Create a new MySQL user (with a secure password).
* Add this user account to the new database and grant it All Privileges.

Write down the database name, the new MySQL username and its password. You will need them soon.

## Step 4: Edit the *wp-config.php* File

Browse to the folder on your local computer where you downloaded your website files to. In that folder there is a file called *wp-config.php* that controls the access between WordPress and your database. Make a copy of this file and store it in another folder on your local computer. This is necessary for restoring the changes we are about to make should something go wrong later. Open the original version of the file with your favorite text editor and make the following three changes:

### Change The Database Name :- define(‘DB\_NAME’,’db\_name’);

### Change the Database Username :- define(‘DB\_USER’,’db\_user’);

### Change The Database User Password :- define(‘DB\_PASSWORD’,’db\_pass’);

## Step 5: Import Your Database

Now that you have a new database to work with we can begin the import process. Launch phpMyAdmin from the cPanel software on your new server and select your new database from the list on the left hands sidebar. Once it opens select the *Import* tab from the navigation menu. In the *File to Import* section click the *Choose File* button and select the SQL file you exported previously. Un-tick the *Partial Import* check box, make sure the format is set to *SQL* and then click the *Go* button. The database import will now begin. The time this import takes varies depending on the size of your database. You should receive a message informing you of the success of the import when it has finished.

## Step 6: Upload The WordPress Files To Your New Host

Now that you have the new database prepared and you’ve reconfigured the wp-config.php file, it is time to begin uploading your website’s files.Connect to your new web host using your FTP program and browse to the folder that your website is going to be held. If this is the primary, or only site being installed on this web server then uploading the files to the public\_html folder is the usual directory.With the remote directory selected you can upload your website files that should now include the updated version of wp-config.php. As with the earlier download, this process can take some time. Don’t delete these files from your local computer once the upload finishes. They are still needed until the final steps have been completed.

## ****MAP All THE LINKS ON THE MYSQL DATABASE TO THE NEW LOCATION ON AWS****

1. **Creating Mapping Rules in the AWS Schema Conversion Tool (AWS SCT)**

Before you convert your schema with AWS SCT, you can set up rules that change the data type of columns, move objects from one schema to another, and change the names of objects. For example, if you have a set of tables in your source schema named test\_TABLE\_NAME, you can set up a rule that changes the prefix test\_ to the prefix demo\_ in the target schema.

You can create mapping rules that perform the following tasks:

Change data type

Move objects

Rename objects

Prefix - add prefix, remove prefix, replace prefix

Suffix - add suffix, remove suffix, replace suffix

You can create mapping rules for the following objects:

Database

Schema

Table

Column

### **Creating Mapping Rules**

You can create mapping rules and save the rules as part of your project. With your project open, use the following procedure to create mapping rules.

**To create mapping rules**

* Choose Mapping Rules from the Settings menu. The Mapping Rules dialog box appears.
* Choose Add new rule. A new row is added to the list of rules.
* Choose the edit icon to configure your rule.
  + For Name, type a name for your rule.
  + For For, choose the type of object that the rule applies to.
  + For where, type a filter to apply to objects before applying the mapping rule. The where clause is evaluated by using a like clause. You can enter an exact name to select one object, or you can enter a pattern to select multiple objects.The fields available for the where clause are different depending on the type of the object. For example, if the object type is schema there is only one field available, for the schema name.
  + For Actions, choose the type of mapping rule you want to create.
  + Depending on the rule type, type one or two additional values. For example, to rename an object, type the new name of the object. To replace a prefix, type the old prefix and the new prefix

.

* After you have configured your mapping rule, choose Save to save your rule. You can also choose Cancel to cancel your changes.
* After you are done adding, editing, and deleting rules, choose Save All to save all your changes.
* Choose Close to close the Mapping Rules dialog box.

### **Viewing Mapping Rules for Objects**

After you set up your mapping rules, you can view the effect of the rules on specific objects in your schema before you convert your schema. In the source schema tree, choose the object you are interested in. In the main view, choose the Mapping tab. The Mapping tab opens and displays a list of all mapping rules that are applied to the object. You can see the name of the object in the source schema and the new name of the object in the target schema. If you have data type rules, you also see the data type of the column in the source schema and the new data type of the column in the target schema.

### **Exporting Mapping Rules**

If you use AWS Database Migration Service (AWS DMS) to migrate your data from your source database to your target database, you can provide information about your mapping rules to AWS DMS. For more information about tasks,

**To export mapping rules**

1. In the AWS Schema Conversion Tool, in the source schema tree, open the context (right-click) menu and choose Export script for DMS. The save dialog box opens.
2. Browse to the location where you want to save your script, and then choose Save. Your mapping rules are saved as a JSON script that can be consumed by AWS DMS.
3. **Converting your Schema By using the AWS Schema Conversion Tool**

After you have connected your project to both your source database and your target Amazon RDS DB instance, your AWS Schema Conversion Tool project displays the schema from your source database in the left panel. The schema is presented in a tree-view format, and each node of the tree is lazy loaded. When you choose a node in the tree view, AWS SCT requests the schema information from your source database at that time.

You can choose schema items from your source database and then convert the schema to equivalent schema for the DB engine of your target DB instance. You can choose any schema item from your source database to convert. If the schema item that you choose depends on a parent item, then AWS SCT also generates the schema for the parent item. For example, if you choose a column from a table to convert, then AWS SCT generates the schema for the column, the table that the column is in, and the database that the table is in.