**Lab Steps**

Task 1: Sign in to AWS Management Console

1. Click on the  button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

**Note :**If you face any issues, please go through [**FAQs and Troubleshooting for Labs**](https://www.whizlabs.com/labs/support-document/faqs-and-troubleshooting).

Task 2: Launch New Ubuntu EC2 Instance

1. Make sure you are in the **US East (N. Virginia)**Region.
2. Navigate to menu in the top, then click on **EC2** in the **Compute** section.
3. Click on **Instances** from the left side bar and then click on 
4. Name : Enter ***SourceEC2Instance***

Table

Description automatically generated with medium confidence

1. **For Amazon Machine Image (AMI):** Select Ubuntu in quickstart and  in the drop box and select **Ubuntu Server 18.04**

Graphical user interface, text, application

Description automatically generated

**Note: If you do not select the mentioned version, you will run into errors while executing few commands.**

1. For Instance Type: select ***t2.micro***

***Graphical user interface, application

Description automatically generated***

1. **For Key pair:** Select **Create a new key pair** Button
   * Key pair name: **WhizKey**
   * Key pair type: **RSA**
   * Private key file format: **.pem**.
2. In Network Settings Click on **Edit**:

* Auto-assign public IP: **Enable**
* Select **Create a new security group**
* Security group name    : Enter ***Migration-SG***
* Description        : Enter ***Migration-SG***

***Graphical user interface, text, application, email

Description automatically generated***

* To add **SSH**,
  + Choose Type    : 
  + Source        : 
* For **MYSQL/Aurora,**
  + Click on **Add security group rule**
  + **Source        : **
  + **Choose Type    : MYSQL/Aurora**

1. Keep Rest thing Default and Click on**Launch Instance**Button.
2. Click on **View Instances**. After 1-2 Mins **Instance State** will become **running**and is ready.
3. **Note down the IPv4 Public IP address - Example: 54.221.120.52**

Task 3: SSH into the EC2 Instance

* Please follow the steps in [SSH into EC2 Instance](https://www.whizlabs.com/labs/support-document/ssh-into-ec-instance).
* **Note:** While SSH into the EC2 use **ubuntu**instead of **ec2-user** because it is a Ubuntu-based OS.

Task 4: Install and Configure MySQL Server

1. Once connected to the server switch to root user:
   * sudo su
2. To update all the packages:
   * apt-get update -y
3. Install MySQL:
   * apt-get install mysql-server -y
4. Login to the MySQL:
   * mysql -u root -p

****

1. Enter Password: **by default, the password is empty.** Press**Enter** to **proceed further.**
2. Enter

exit;

1. Now, we have to set the password for the root user. Please keep a note of this password as we will use this in upcoming steps. To set the default password,  please run the below-provided commands one by one:
   * We must first stop the MySQL service using the command :

/etc/init.d/mysql stop

* + Create a directory using the command :

mkdir /var/run/mysqld

* + Change the ownership of the directory using :

chown mysql /var/run/mysqld

* + Now start MySQL using :

mysqld\_safe --skip-grant-tables&

**Text

Description automatically generated**

* + You will be seeing something similar to the above picture, then Press [**Enter**].
  + Connect to MySQL with root user using :

mysql -u root

* + Enter :

UPDATE mysql.user SET authentication\_string=null WHERE User='root';

* + Enter :

FLUSH PRIVILEGES;

* + Enter :

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'source123';

* + Enter :

FLUSH PRIVILEGES;

* + Enter :

exit;

* + Now we must stop all services before starting again by using :

sudo killall -u mysql

****

* + You will see something similar to the above picture, hit [**Enter**] if you do not come out of the process message.
  + Start the service using :

/etc/init.d/mysql start

* + Log in to MySQL again using :

mysql -p -u root

* + Password: ***source123***

1. Now you can only log in using the password you have set and nothing else.
2. After successful login, you will be able to see the welcome message and **mysql>** prompt waiting for your command:

**Text

Description automatically generated**

1. To **exit** out of **MySQL,** run the following command**: exit;**
2. Make sure you do not give any extra spaces while executing these commands.

Task 5: Update the MySQL configuration file

We will open **mysqld.cnf**file and check the below details to configure them. If any rule is missing, then we have to manually add those in the mysql config file. To edit that file, follow the below steps in Source Instance.

1. To edit the mysql config file use the below provided command:
   * nano /etc/mysql/mysql.conf.d/mysqld.cnf
2. Check the below details already in the MySQL configuration file.
   * bind-address = 0.0.0.0
   * server-id = 1
   * log\_bin=mysql-bin
   * binlog\_format=row
3. If any rule is not available, then enter the missed ones manually and save the file**.**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

1. To save the file: Press ctrl+x and confirm by pressing ‘y’ and hit enter.
2. Restart the Mysql server using the below command:
   * /etc/init.d/mysql restart
3. Source Database is configured.

Task 6: Create an Amazon RDS Database

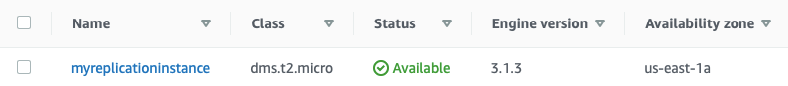
1. Click on the  and select the RDS under the **Database** section.
2. In the left navigation pane, click on **Databases**.
3. Make sure you are in **N.Virginia** Region.
4. Click 
5. **Specify DB Details**:

* Instance specifications
  + Database creation method : **Standard create**
  + Engine options           : Select **MySQL**
  + Version                       : **Default**
  + Templates                   : Select **Free tier**
  + DB instance identifier    : Enter ***mydbinstance***
  + Master username.         : Enter***awsrdsuser***
  + Master password and Confirm password: Enter ***whizlabs123***  
    **Note:**This is the username/password combo used to log onto your database. Please make note of them somewhere safe.
  + DB instance class        : Select **Burstable classes db.t2.micro — 1 vCPUs, 1 GiB RAM**
  + **NOTE : Select Include previous generation classes If t2.micro is not visible**
  + Storage type            : Select **General Purpose SSD (gp2)**
  + Allocated storage        : Select **20 (default)**
  + Enable storage autoscaling     : **Uncheck**
  + Vitual Private Cloud(VPC) : Select **Default VPC**
  + Subnet group : Select **Default**
  + Public Access : Select **No**
  + VPC Security groups : Select **Choose existing**
  + Existing VPC security group name : Remove the **default**security group and select **Migration-SG**from the dropdown list
  + Scroll down to **Additional Configuration** options
    - Initial database name: Enter ***myrdsdatabase***
    - DB parameter group: Select **default**
    - Option group: Select **default**
    - Enable automated backups: **uncheck**
    - Enable Enhanced monitoring: **uncheck**
    - Enable auto minor version upgrade: **uncheck**
    - Maintenance window: Select **No preference**
    - Enable deletion protection: **uncheck**

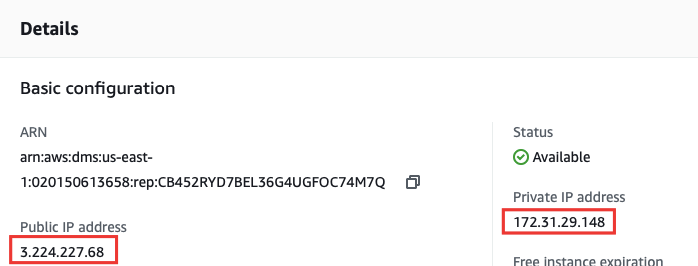
1. Leave other parameters as default.Scroll to the bottom of the page, Click **Create database**.
2. It will take around 5 minutes for the database to become available. Once the status changes from **creating** to the **available**, the database is ready.
3. Open **mydbinstance**and note down the **Endpoint** of RDS under **Connectivity and security**
   * **Example: mydbinstance.c81x4bxxayay.us-east-1.rds.amazonaws.com**

Task 7: Create a Replication instance in DMS

1. Click on**** and then choose Database Migration Service under the **Migration & Transfer.**
2. Make sure you are in the **N.Virginia** region.
3. Click on ****
4. In the **Replication instance configuration**section,
   * Name                             : Enter ***myreplicationinstance***
   * Description                     : Enter ***To replicate EC2-Mysql to AWS RDS***
   * Instance class                : Select **dms.t3.micro**
   * Engine version               : Default
   * Allocated storage (GB)   : **10 GB**
   * VPC                                : default VPC
   * Multi AZ                           :  Select **Dev or test workload (Single-AZ)**
   * Publicly accessible          : **Check**
5. In**Advanced security and network configuration** section,
   * Replication subnet group  : Default
   * Availability zone                  : Default
   * VPC security group(s)        : Enter ***Migration-SG***
   * KMS master key                  : Default
6. Leave other settings as default.
7. Click on the  **Create** button to create the replication instance.
8. It will take 5 minutes for replication instance to be created. Once status changes to **available**, click on the instance and scroll down. You will find the details section of the replication instance.



1. Click on the **myreplicationinstance** from **Details** section, copy the private & public IP address and note it down on the notepad.



* Public IP address ?**3.224.227.68**
* Private IP address ?**172.31.29.148**

Task 8: Configure Replication Instance details in Source EC2 Instance

1. SSH back into the Source EC2 instance. For more details go through **SSH into Source Database EC2 instance from Mac or Windows systems**.
2. Switch to root user:
   * sudo su
3. Login to the MySQL:
   * Enter:

mysql -u root -p

* + Press [**Enter**]
  + Enter password: ***source123***
  + Press [**Enter**]

1. We need to grant root access to the replication instance to connect with the MySQL server on Source EC2. To give access, follow below steps:
   * Command Syntax:
     + **GRANT ALL ON \*.\* TO root@'***<<Private IP of Replication Instance>>***' IDENTIFIED BY '***your-root-password***';**
     + **Example:GRANT ALL ON \*.\* TO root@'172.31.29.148' IDENTIFIED BY 'source123';**
2. And repeat the same step now with the **Public Ip address** of the replication instance.
   * **GRANT ALL ON \*.\* TO root@'3.224.227.68' IDENTIFIED BY '*source123*';**
3. Save the changes by using the following command:
   * FLUSH PRIVILEGES;
4. Enter **exit**to Exit MySQL and to restart the MySQL server, run the below command.
   * /etc/init.d/mysql restart
5. Replication instance has access for Source Instance MySQL Database.

Task 9: Create Endpoints in DMS

We have to create the Source and Target endpoints for EC2 and RDS Instances. These endpoints will help to connect replication instance with both source and target machines.

1. Source Endpoint

* Make sure you are in **N.Virginia** **(us-east-1)** region.
* To create an Endpoint, Click on the Endpoints(Left panel) in the DMS service console page and click on the 
* Follow the below steps to complete**Endpoint type:**
  + Select endpoint as ****
  + Select RDS DB instance: Uncheck (**This is for Source i.e Mysql on EC2**)

2. Endpoint configuration:

* Endpoint identifier     : Enter ***sourcemysqlendpoint***
* Descriptive Amazon Resource Name (ARN): ***sourcemysqlendpoint***
* Source engine           : Select **MySQL**
* Access to endpoint database: Choose **Provide access information manually**
* Server name               :**54.221.120.52**- Public IP address of Source EC2 Instance(Enter your IP)
* Port                             : Enter ***3306***
* Secure Socket Layer (SSL) mode: **None**
* User name                 : Enter ***root***
* Password                   : Enter ***source123***

3. Leave other configurations as default.

4. Test endpoint connection:

* VPC                                   : Default
* Replication instance          : Enter ***myreplicationinstance***(Which was created earlier)
* Click on **Run test** to test the connection. If all are working fine, you will be able to see the status as “successful” as shown in the below screenshot.

Graphical user interface, text, application, email

Description automatically generated

* Click on **Create Endpoint**.

Task 10: Target Endpoint:

1. To create an Endpoint, Click on the Endpoints(Left panel) in the DMS service console page and click on the 
2. Follow the below steps to complete**Endpoint type:**
   * Select endpoint as ****
   * Select RDS DB instance: check (**This is for Target i.eRDS Instance**)
   * Select RDS Database: ***mydbinstance***
3. **Endpoint configuration:**
   * Endpoint identifier    : Enter ***mydbinstance***
   * Descriptive Amazon Resource Name (ARN): Enter ***awsrdsendpoint***
   * Target engine            : Enter**MySQL**
   * Access to endpoint database: Choose **Provide access information manually**
   * Server name             : Enter ***mydbinstance.c81x4bxxayay.us-east-1.rds.amazonaws.com* ? DNS Endpoint of RDS database**
   * Port                             : Enter ***3306***
   * Secure Socket Layer (SSL) mode: None
   * User name                 : Enter ***awsrdsuser***
   * Password                   : Enter ***whizlabs123***
4. Leave other settings as default.
5. **Test endpoint connection:**
   * VPC                                   : Default
   * Replication instance          : Enter ***myreplicationinstance***(Which we created earlier)
   * Click on **Run test** to test the connection.Graphical user interface, text, application, email

     Description automatically generated
6. Click on the **Create endpoint** button.

Task 11: Create a simple custom Database on Source EC2:

Create a simple database and create a table inside which will be migrated using DMS.

1. SSH back to Source EC2 Instance.
2. Connect to Source MySQL Client

* mysql -u root -p
* Enter Password : ***source123***

1. Create a Database

* CREATE DATABASE SchoolDB;

****

* View the database created
  + show databases;

**Text

Description automatically generated**

1. Switch the database **SchoolDB.**
   * use SchoolDB;

****

1. Create a sample Table of **students.**

|  |
| --- |
| CREATE TABLE students (  subject\_id INT AUTO\_INCREMENT,  subject\_name VARCHAR(255) NOT NULL,  teacher VARCHAR(255),  start\_date DATE,  lesson TEXT,  PRIMARY KEY (subject\_id)  ); |

**Text

Description automatically generated**

1. See the **student** table.
   * show tables;

Graphical user interface, text

Description automatically generated with medium confidence.

1. Insert data into the table

|  |
| --- |
| INSERT INTO students(subject\_name, teacher) VALUES ('English', 'John Taylor');  INSERT INTO students(subject\_name, teacher) VALUES ('Science', 'Mary Smith');  INSERT INTO students(subject\_name, teacher) VALUES ('Maths', 'Ted Miller');  INSERT INTO students(subject\_name, teacher) VALUES ('Arts', 'Suzan Carpenter'); |

1. Check the items added in the Table
   * select \* from students;

**Graphical user interface, text

Description automatically generated**

1. After database migration, this new custom table can be used as proof of database migration.

Task 12: Checking AWS RDS Database before Migration

First, check the databases and tables that exist on the AWS RDS Instance. So that after migration, you will be able to find the new changes. We can use the existing Source EC2 Instance to connect to AWS RDS.

1. SSH into the Source EC2 instance. For more details go through **SSH into EC2 instance from Mac or Windows systems**.
2. Switch to root user:

sudo su

1. Now, connect to the Amazon RDS Instance by running below command:
   * Syntax :**mysql -h <RDS Instnace Endpoint> -u <User Name> -p**
   * **Example:mysql -u mydbinstance.c81x4bxxayay.us-east-1.rds.amazonaws.com -u awsrdsuser -p**
   * Enter Password: ***whizlabs123***
2. After successful login, run the below command.
   * Show databases;

**Text

Description automatically generated**

1. A database by name ***SchoolDB*** does not exists now. After migration SchoolDB database will be available here.

Task 13: Create a Database Migration Task

An AWS Database Migration Service task is where all the migration process happens. We will specify the tables and schemas to use for the migration and any special processing, such as logging requirements, control table data, and error handling.

1. Navigate to AWS DMS console and click on the **Database migration tasks**.
2. Make sure you are in **N.Virginia** **(us east-1)** Region.
3. Click on the ****.
4. Create a database migration task:
   * **Task configuration:**
     + Task identifier                         : ***Database-Migration-Task***
     + Replication instance               : Select **myreplicationinstance**
     + Source database endpoint     : Enter ***sourcemysqlendpoint***
     + Target database endpoint       : Enter ***mydbinstance***
     + Migration type                         : **Migrate existing data and replicate ongoing changes**
   * **Task settings:**
     + Editing mode: **Wizard**
     + Target table preparation mode           : **Do nothing**
     + Include LOB columns in replication: **Limited LOB mode**
     + Maximum LOB size (KB): **32 KB**
     + Enable validation: **Uncheck**
     + Enable Cloudwatch Logs: **Uncheck**
     + **Leave Advance tasks settings as default.**
   * Table mappings:
     + Editing mode         : **Wizard**
     + Selection rules        : 
     + Schema        : Select **Enter a Schema**
     + Schema name        : Enter ***%SchoolDB***(Database name)
     + Table name             : Enter ***%students*** (Table name)
     + Action                      : **Include**
   * Leave other settings as default.
5. Click on **Create task**.
6. Now the migration of Database will occur. Usually it will take around 1 minute in this lab.

Task 14: Status of AWS Database Migration Tasks:

1. Navigate toin left panel of DMS page.
2. If you followed all the previous steps correctly, it will show the migration task status as
3. Now the Migration of database to Amazon RDS is completed.
4. To check the migration status and details, we need to connect to destination MySQL database which is in AWS RDS Instance.
5. SSH back into Source EC2 Instance.
6. Connect to AWS RDS instance.
   * mysql -u mydbinstance.c81x4bxxayay.us-east-1.rds.amazonaws.com -u awsrdsuser -p
   * Enter Password: ***whizlabs123***
7. Once inside MySQL Client, check the databases available.
   * show databases;

Text

Description automatically generated

1. Run command:
   * use SchoolDB;

**Text

Description automatically generated**

1. Run command:
   * show tables;

Timeline

Description automatically generated

1. Run command:
   * select \* from students;

Text

Description automatically generated

1. Now we can see the ***SchoolDB*** which was available in Source EC2 Ubuntu Server migrated to Amazon RDS Instance Database.

Task 15: Validation Test

1. Once the lab steps are completed, please click on the A picture containing text

   Description automatically generated button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :

Table

Description automatically generated

**Completion and Conclusion**

1. You have launched an EC2 instance and installed a MySQL server on it.
2. You have created AWS RDS(MySQL) instance and accessed via an EC2 instance.
3. You have learned how to create Replication Instance, Source Endpoint, Destination Endpoint.
4. You have learned how to create Database Migration Task and migrate Database from source database to destination database.
5. You have learned how to Migrate your MySQL database exists on the EC2 instance to AWS RDS Instance.

**End Lab**