Using Amazon RDS for Applications

**SPL-06 - Version 4.2.24**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Lab overview**

This lab demonstrates how to use an Amazon Relational Database Service (Amazon RDS) database with applications. Any application that uses a Structured Query Language (SQL) database such as MySQL, SQL Server, Oracle or PostgreSQL can use Amazon RDS as a scalable, reliable database.

You reconfigure a Drupal Open Source Content Management System (CMS) to use Amazon RDS for MySQL as the backend database with a multi-Availability Zone (AZ) deployment model.

OBJECTIVES

By the end of this lab, you should be able to do the following:

* Launch an Amazon RDS database.
* Transition an application to use the Amazon RDS database.
* Change the Instance Type of an Amazon RDS database.
* Configure an Amazon RDS database for High Availability.

TECHNICAL KNOWLEDGE PREREQUISITES

To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

* **Command:** A command that you must run.
* **Expected output:** A sample output that you can use to verify the output of a command or edited file.
* **Note:** A hint, tip, or important guidance.
* **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
* **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.
* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Consider:** A moment to pause to consider how you might apply a concept in your own environment or to initiate a conversation about the topic at hand.
* **Task complete:** A conclusion or summary point in the lab.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**Warning:** Do not change the **Region** unless instructed.

COMMON SIGN-IN ERRORS

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

SERVICES USED IN THIS LAB

**What is Amazon RDS?**

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, freeing you up to focus on your applications and business.

Amazon RDS gives you access to the capabilities of a familiar MySQL, Oracle, Microsoft SQL Server, or PostgreSQL database engine. This means that the code, applications, and tools you already use today with your existing databases can be used with Amazon RDS. Amazon RDS automatically patches the database software and backs up your database, storing the backups for a user-defined retention period and enabling point-in-time recovery. You benefit from the flexibility of being able to scale the compute resources or storage capacity associated with your Database Instance (DB Instance) via a single API call.

Amazon RDS DB Instances can be provisioned with either standard storage or Provisioned IOPS storage. Amazon RDS Provisioned IOPS is a storage option designed to deliver fast, predictable, and consistent I/O performance, and is optimized for I/O-intensive, transactional (OLTP) database workloads.

In addition, Amazon RDS makes it easy to use replication to enhance availability and reliability for production workloads. Using the **Multi-AZ deployment option** you can run mission critical workloads with high availability and built-in automated fail-over from your primary database to a synchronously replicated secondary database in case of a failure. Amazon RDS for MySQL also enables you to **scale-out** beyond the capacity of a single database deployment for read-heavy database workloads. As with all Amazon Web Services, there are no up-front investments required, and you pay only for the resources you use.

Some common features of Amazon RDS are:

**Pre-configured Parameters:** Amazon RDS DB Instances are pre-configured with a sensible set of parameters and settings appropriate for the DB Instance class you have selected. You can simply launch a MySQL, Oracle or SQL Server DB Instance and connect your application within minutes without additional configuration. If you desire additional control, you can achieve it via DB Parameter Groups.

**Monitoring and Metrics:** Amazon RDS provides Amazon CloudWatch metrics for your DB Instance deployments at no additional charge. You can use the AWS Management Console to view key operational metrics for your DB Instance deployments, including compute/memory/storage capacity utilization, I/O activity, and DB Instance connections.

**Automatic Software Patching:** Amazon RDS makes sure that the relational database software powering your deployment stays up-to-date with the latest patches. You can exert optional control over when and if your DB Instance is patched via DB Engine Version Management.

**Automated Backups:** Turned on by default, the automated backup feature of Amazon RDS enables point-in-time recovery for your DB Instance. Amazon RDS backs up your database and transaction logs and store both for a user-specified retention period. This allows you to restore your DB Instance to any second during your retention period, up to the last five minutes. Your automatic backup retention period can be configured to up to thirty five days.

**DB Snapshots:** DB Snapshots are user-initiated backups of your DB Instance. These full database backups are stored by Amazon RDS until you explicitly delete them. You can create a new DB Instance from a DB Snapshot whenever you desire.

**Provisioned IOPS:** Using the Amazon RDS APIs, or with a few clicks on the AWS Management Console, you can provision the IOPS (Input/Output Operations Per Second) for your database instance and scale it easily. This functionality is available to you in two stages. Starting immediately, when you create new DB Instances using the AWS Management Console or the Amazon RDS APIs, you can provision from 1,000 IOPS to 10,000 IOPS with corresponding storage from 100GB to 1TB for MySQL and Oracle engines. You can start small and scale up in increments of 1,000 IOPS and 100GB of storage. If you are using SQL Server then the maximum IOPS you can provision is 7,000 IOPS.

**Push-Button Scaling:** Using the Amazon RDS APIs or with a few clicks on the AWS Management Console, you can scale the compute and memory resources powering your deployment up or down. Scale compute operations typically complete within a handful of minutes. For MySQL and Oracle database engines, as your storage requirements grow, you can also provision additional storage on-the-fly with zero downtime. If you are using RDS Provisioned IOPS with the MySQL and Oracle database engines, you can also scale the throughput of your DB Instance by specifying the IOPS rate from 1,000 IOPS to 10,000 IOPS and corresponding storage from 100GB and 1TB in 1,000 IOPS and 100GB increments.

**Automatic Host Replacement:** Amazon RDS automatically replaces the compute instance powering your deployment in the event of a hardware failure.

**Replication:** Amazon RDS provides two distinct but complementary replication features: Multi-AZ deployments and Read Replicas that can be used in conjunction to gain enhanced database availability, protect your latest database updates against unplanned outages, and scale beyond the capacity constraints of a single DB Instance for read-heavy database workloads. Multi-AZ deployments are available for the MySQL and Oracle database engines. Read Replicas are currently supported for the MySQL database engine.

**Isolation and Security:** Using Amazon VPC, you can isolate your DB Instances in your own virtual network, and connect to your existing IT infrastructure using industry-standard encrypted IPsec VPN. The VPC functionality is supported by all RDS DB Engines. To learn more about Amazon RDS in VPC, refer to the Amazon RDS User Guide. In addition, using Amazon RDS, you can configure firewall settings and control network access to your DB Instances.

AWS SERVICES NOT USED IN THIS LAB

AWS service capabilities used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

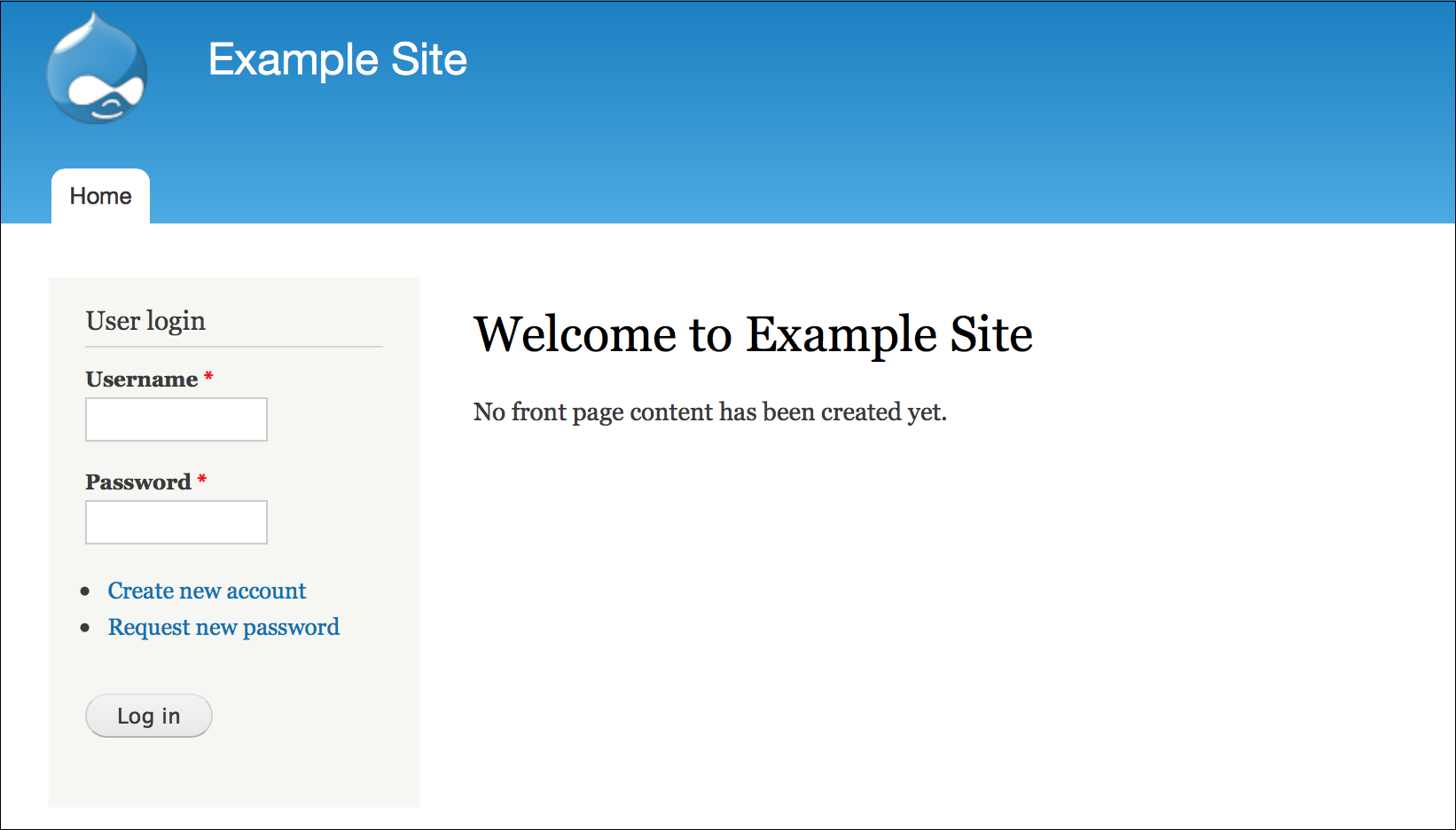
**Task 1: Access the Content Management System (CMS)**

An Amazon EC2 instance has been automatically created for you. It has been configured as a fully-functional CMS application, using its own local database (not Amazon RDS).

In this task, you access the application and store some data in the database.

1. Copy the value of **PublicIP** found to the left of these instructions, and paste it into a new browser tab.

You should see a webpage similar to this:



*Image description: The preceding diagram depicts the web page for an Example Site with a fully-functional CMS application.*

1. Log into the CMS using:

* **Username:**

admin

* **Password:**

admin

You should be taken to the **Example Site** home page, where you can create a new page that gets stored in the local database.

1. Choose **Add content**.
2. Choose **Article**, then configure:

* **Title:**

My First Article

* **Body:** *Write some text here*
* Scroll to the bottom, then choose **Save**

Your article is saved in the database and should appear on the front page of the CMS application.

**Task complete:** You have successfully accessed the application and stored some data in the database.

**Task 2: Launch an Amazon RDS for MySQL Database**

To make the application more scalable and durable, you migrate your data from your local database to an Amazon RDS database. In this task, you launch an **Amazon Relational Database Service (RDS) for MySQL** database.

1. Without closing the CMS browser tab, return to the web browser tab that contains the AWS Management Console. (If you can’t find it, choose the **Open Console** button in these instructions.)
2. At the top of the AWS Management Console, in the search bar, search for and choose

RDS

.

1. In the left navigation pane, choose **Databases**.
2. Choose **Create database** then configure:
3. In the **Engine options** section, for **Engine type**, choose  **MySQL**.
4. In the **Templates** section, choose  **Dev/Test**.
5. In the **Availability and durability** section, choose  **Single DB instance**.
6. In the Settings section, configure:

* For **DB instance identifier**, enter

DrupalLab

.

* For **Master username**, enter

root

.

* For **Credentials management**, choose **Self managed**.
* For **Master password**, enter

admin123

.

* For **Confirm master password**, enter

admin123

.

1. In the **Instance configuration** section, configure:

* For **DB instance class**, choose **Burstable classes (includes t classes)**.
  + Use the dropdown menu and select **db.t3.small**.

1. In the **Connectivity** section, configure:

* For **Virtual Private Cloud (VPC)**, use the dropdown menu and select **Lab VPC**.
* For **Public access**, choose  **No**.
* For **VPC security group (firewall)**, choose  **Choose existing**.
* For **Existing VPC Security groups**:
  + Use the dropdown menu and select **DBSecurityGroup** security group.
  + Remove **default** security group.

The DBSecurityGroup was created for you and permits inbound traffic from the Web Server to your Amazon RDS database.

1. In the **Monitoring** section:

* De-select  **Enable Enhanced Monitoring**.

1. Expand the  **Additional configuration** section *(near the bottom of the page)* then:

* For **Initial database name**, enter

drupaldb

.

* De-select  **Enable automated backups**.

**Consider:** You are disabling database backups in this *lab* so that you can progress through the lab without having to wait for a backup to complete. However, in the real-world you want database backups enabled.

1. Scroll to the bottom of the screen and choose **Create database**.

This screen displays the details of your database that is launching. On this screen, you can monitor the progress of your DB Instance.

If you see a pop-up window titled **Suggested add-ons for drupallab**, choose **Close**.

The database takes several minutes to launch. However, you can **continue with the next task** while you are waiting for the database to become available.

**Task complete:** You have successfully launched an Amazon Relational Database Service (RDS) for MySQL database.

**Task 3: Login to your Amazon EC2 Instance and Backup the CMS Database**

In this task, you login to your Amazon EC2 instance running the CMS so that you can modify the application configuration. You then backup the contents of the existing CMS database to a file so that it can be copied to the RDS DB Instance.

1. Copy the value of **WebServerSessionUrl** found to the left of these instructions and paste it into a new browser tab to connect to it.

This connects to you to your EC2 instance using Session Manager.

1. **Command:** Run the following command in your Session Manager session:

mysqldump -u root -padmin123 drupaldb > backup.sql

This command creates a backup of your existing database and saves the backup to the *backup.sql* file.

1. **Command:** Run the following command in your Session Manager session:

tail backup.sql

Using this command, you can verify that the backup was completed successfully.

**Expected output:** You should see **Dump completed** in the screen output.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-06-28 16:23:54

**Task complete:** You have successfully logged in to your Amazon EC2 instance running the CMS application, modified the application configuration, and backed up the contents of the existing CMS database to a file so that it can be copied to the RDS DB Instance.

**Task 4: Transition the Drupal CMS Instance to use the RDS database**

In this task, you copy the database contents from the backup you made earlier and then reconfigure the CMS to use the RDS DB Instance.

1. Return to the web browser tab showing the Amazon RDS Management Console.
2. In the left navigation pane, choose **Databases**.
3. Verify that the **Status** of the **drupallab** database is  Available . If it isn’t, wait for it to become so.

**Refresh:** You can refresh the page to update the status.

1. Choose **drupallab**.
2. Under the **Connectivity & security** tab, copy the **Endpoint** to a text editor.

The **Endpoint** should look similar to: *drupallab.c617fmllbu1n.us-west-2.rds.amazonaws.com*

You use this endpoint in a moment to configure the CMS application.

1. **Copy edit:** Copy the following command and paste it into the text editor (do not run it yet!).

mysql --user=root --password=admin123 --database=drupaldb --host=ENDPOINT < backup.sql

1. Replace **ENDPOINT** in the copied command with the endpoint you copied earlier.

Your command text should now look similar to: *mysql --user=root --password=admin123 --database=drupaldb --host=drupallab.cepdufdx0foh.us-west-2.rds.amazonaws.com < backup.sql*

1. Copy and paste the updated command to your Session Manager session and run it.

This copies the backup database schema and data to the new RDS DB Instance.

Next, you need to reconfigure the CMS to use the new RDS DB Instance.

1. **Command:** Open the *settings.php* file using your favorite command line text editor. For example, to use the **vim** editor in your Session Manager session, run the command:

sudo vim /var/www/html/sites/default/settings.php

1. Jump to the correct section:

* Press **/**
* Type

admin123

* Press **Enter**

You should see a section similar to the following:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$databases = array (

'default' =>

array (

'default' =>

array (

'database' => 'drupaldb',

'username' => 'root',

'password' => 'admin123',

'host' => 'localhost',

'port' => '',

'driver' => 'mysql',

'prefix' => 'drupal\_',

),

),

);

1. Replace **localhost** with the RDS **Endpoint** value you copied earlier.

* Use the up, down, right, and or left arrow keys on your keyboard to move the cursor before **localhost**.
* Enter *Insert Mode* to edit by pressing **i**.
* Paste in the **Endpoint** value.
* Press **Esc** key to exit *Insert Mode*.
* Move the cursor to the first letter of **localhost**.
* Press **x** repeatedly (nine times, once for each letter of **localhost**).

Your updated file should look similar to:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\* EXAMPLE OUTPUT \*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$databases = array (

'default' =>

array (

'default' =>

array (

'database' => 'drupaldb',

'username' => 'root',

'password' => 'admin123',

'host' => 'drupallab.ch50kbdft1dg.us-east-1.rds.amazonaws.com',

'port' => '',

'driver' => 'mysql',

'prefix' => 'drupal\_',

),

),

);

1. To save your file:

* Enter the colon character **“:”** (Press **Shift** + **:**).
* The character **“:”** appears at the bottom of the editor.
* Type **wq!**.
* Press **Enter**.

The config file is now saved with the new RDS endpoint. The CMS now uses the RDS database for storing its data.

You can now confirm that the database content has been successfully transferred over to the RDS instance.

1. Return to the web browser tab with the CMS application. (It’s the browser tab you used in task 1, with “Example Site” in the tab name. If you cannot find it, copy the *WebServerSessionUrl* from the left of these instructions and paste it into a new browser tab.)
2. **Refresh:** Refresh the web page.

You should see the article that you created earlier. This confirms that the database backup was successfully copied to the RDS DB Instance.

**Caution:** If you received an application error when refreshing the web page, check the previous instructions to confirm that you correctly edited the configuration file.

**Task complete:** You have successfully copied the database contents from the backup you made earlier and then reconfigured the CMS to use the RDS DB Instance.

**Task 5: Change the RDS Instance Size**

Scaling an Amazon RDS database is simple via the AWS Management Console. You can easily grow the database or change the underlying server size.

In this task, you convert the database to run on a larger instance type and configure it to support **Multi-AZ**.

1. In the **RDS Management Console**, in left navigation pane, choose **Databases**, then:

* Select the radio button next to **drupallab**.
* Choose **Modify**.

1. In the **Instance configuration** section, configure:

* For **DB instance class**, choose  **Burstable classes (includes t classes)**.
* Use the dropdown menu and select **db.t3.medium**.

1. In the **Availability & durability** section, configure:

* For **Multi-AZ deployment**, choose  **Create a standby instance (recommended for production usage)**.

This option creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. This distributes the database across multiple Availability Zones.

1. Scroll to the bottom of the page, then choose **Continue**.
2. In the **Schedule modifications** section, for **When to apply modifications**, choose  **Apply immediately**.
3. Choose **Modify DB instance**.

This change takes approximately **10-15 minutes** to complete.

1. In the **RDS Management Console**, in left navigation pane, choose **Databases**, then refresh the screen.

**Refresh:** You can choose refresh in the upper right-hand corner to occasionally update the status.

**Consider:** What if the original Availability Zone you selected somehow becomes unavailable? A rare occurrence, but certainly possible. By selecting the Multi-AZ Deployment option in this latest modification, you have made the database highly available, and reads/writes to the database in secondary AZ remain functional.

The following modifications are currently being performed:

* A new **Master database** is being created.
* The **Instance Type** for the database is the larger *db.t3.medium* size, which has more Memory.
* Amazon RDS automatically detaches the **disk storage** from the existing database and reattach it to the new Master database.
* Amazon RDS also creates a **new Secondary database** in a different Availability Zone (which is in a different physical data center). The data in the Master database should be automatically replicated to the Secondary database.

You are creating a very durable database, and you can even perform functions such as backups without incurring downtime because the backup is performed using the Secondary copy of the database.

Behind the scenes, Amazon RDS replicates data synchronously and your application uses the Master instance for all database operations. Should the Master instance fail, RDS should automatically *promote* the Secondary instance to be the new Master while automatically replacing the failed server. Your connection Endpoint remains the same, so there should be minimal impact on your application.

1. Verify that the **Status** of the **drupallab** database is  Available . If it isn’t, wait for it to become so.
2. Choose **drupallab**.
3. Choose **Logs & events** tab.

* In the **Recent events** section, you see the changes being applied to the database.

1. Return to your web browser where your CMS application is running. Use the application as you would normally – navigate through the links or create new entries. The data is now being stored in a larger, Multi-AZ Amazon RDS database.

**Note:** If you encounter errors using the CMS application at this point, try closing the tab and loading the original **PublicIP** in a new browser tab.

**Task complete:** You have successfully converted the database to run on a larger instance type and configured it to support Multi-AZ.

**Conclusion**

You have successfully done the following:

* Launched an Amazon RDS database.
* Transitioned an application to use the Amazon RDS database.
* Changed the Instance Type of an Amazon RDS database.
* Configured an Amazon RDS database for High Availability.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional resources**

* [Amazon Relational Database Service](http://aws.amazon.com/rds)
* [Drupal](https://www.drupal.org/)

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).