



Using Amazon RDS for Applications

Table of Contents

Introduction	3
Amazon Relational Database Service (Amazon RDS)	3
Start your <i>qwikLAB</i> ™	6
AWS Management Console.....	8
Confirm your AWS Region	8
Configure the CMS	11
Log into the Drupal CMS instance using SSH	13
Connect to your EC2 Instance via SSH (Windows)	14
Download PuTTY	14
Download your EC2 Key Pair private key file	14
Connect to the EC2 Instance using SSH and PuTTY.....	15
Connect to your EC2 Instance via SSH (OS X and Linux).....	17
Download your EC2 Key Pair private key file	17
Connect to the EC2 Instance using the OpenSSH CLI client.....	17
Backup the CMS Database	18
Backup	18
Integrate Drupal CMS Application with Amazon RDS.....	19
Determine in which Availability Zone your instance is running.....	19
Set Up a Single-AZ Amazon RDS Database	19
Add Your EC2 Security Group to Database List.....	24
Transition the Drupal CMS Instance to use the Amazon RDS database.....	24
Import the data from the backup that we performed earlier.....	24
Reconfiguring the settings.php file to point to the new Amazon RDS database	24
Change the Amazon RDS Instance Size.....	27
Conclusion	28
End Lab	29

Copyright © 2013 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.

Introduction

In this lab session, we are going to launch a Drupal Open Source CMS with Amazon RDS for MySQL as the backend database with a multi-AZ deployment model.

Amazon Relational Database Service (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 or Microsoft SQL Server 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.

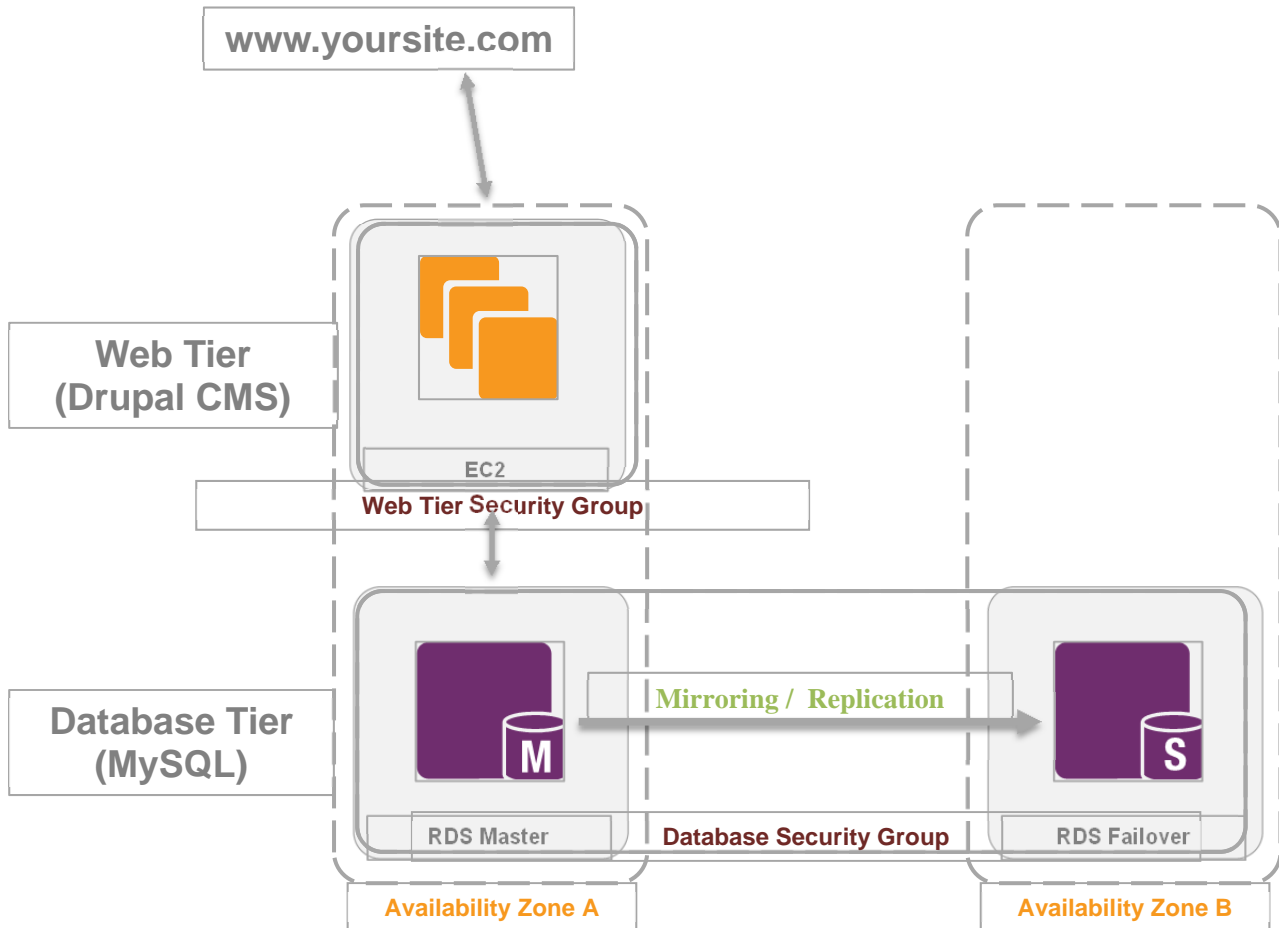
The features provided by Amazon RDS depend on the DB Engine you select. Visit the Amazon RDS for MySQL page (<http://aws.amazon.com/rds/mysql/#features>) for supported features for the MySQL engine, the Amazon RDS for Oracle Database page (<http://aws.amazon.com/rds/oracle/#features>) for supported features of the Oracle database engine or the Amazon RDS for SQL Server page (<http://aws.amazon.com/rds/sqlserver/#features>) for supported features for the SQL Server engine.

- **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 will make 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 will backup 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 will be 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 Amazon 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 will automatically replace 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 (<http://aws.amazon.com/rds/mysql/#Multi-AZ>) and Oracle (<http://aws.amazon.com/rds/oracle/#Multi-AZ>) database engines. Read Replicas (http://aws.amazon.com/rds/mysql/#Read_Replica) are currently supported for the MySQL database engine.
- **Isolation and Security**– Using Amazon VPC (<http://aws.amazon.com/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 Amazon RDS DB Engines. To learn more about Amazon RDS in VPC, refer to the Amazon RDS User Guide (http://docs.amazonwebservices.com/AmazonRDS/latest/UserGuide/USER_VPC.html). In addition, using Amazon RDS, you can configure firewall settings and control network access to your DB Instances.

The diagram below is what we will build during the lab. Before looking at the details, let's be clear that in an hour the result will be incomplete; yet what you learn here will allow you to create a more complete solution "back home".



This training is intended for application administrators & development DBA's who have a combination of needs such as:

- Run existing or new applications, code, or tools that require a relational database
- Want native access to a MySQL, Oracle or SQL Server database engines, but prefer to offload the infrastructure management and database administration to AWS
- Want to exploit the Multi-AZ and Read Replica features (currently available for MySQL only) to achieve enhanced database availability and read scalability
- Like the flexibility of being able to scale their database compute and storage resources with an API call, and only pay for the infrastructure resources they actually consume

Start your *qwikLAB*™

1. Start your *qwikLAB*™

Use the 'Start Lab' button to start your lab.

(Hint: If you are prompted for a token, please use one you've been given or have purchased.)



You will see the lab creation in progress.

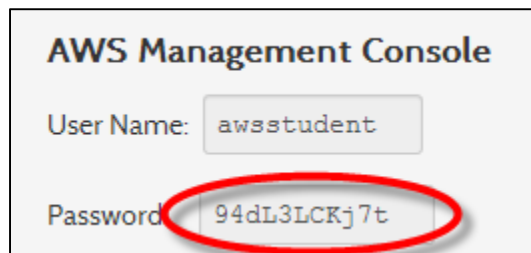


2. Note a few properties of the lab.

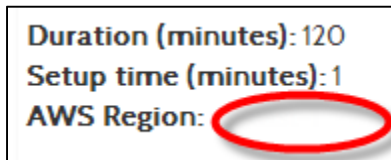
- a. **Duration** - The time the lab will run for before shutting itself down.
- b. **Setup Time** - The estimated lab creation time on starting the lab.
- c. **AWS Region** - The AWS Region the lab resources are being created in.

3. Copy the Password provided.

- d. Hint: selecting the value shown and using Ctrl+C works best



4. Note the AWS Region set for your lab in *qwikLAB*™



5. Click the 'Open Console' button.

Open Console

6. Make sure that you are not logged into any other instances of the AWS console (in a student account or your own account), as this may cause conflicts when you open the console and log in below for this lab.
7. Login to the AWS Management Console

Enter the User Name '**awsstudent**' and paste the password you copied from the lab details in *qwikLAB™* into the Password field.

Click on the 'Sign in using our secure server' button.

In this step you logged into the AWS Management Console using login credentials for a user provisioned via AWS Identity Access Management in an AWS account by *qwikLAB™*.

Amazon Web Services Sign In

Please enter the AWS Identity & Access Management (IAM) User name and password assigned by your system administrator to sign in.

AWS Account: 832809622232

User Name:

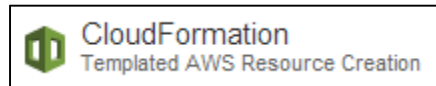
Password:

Please contact your system administrator if you have forgotten your user credentials.

[Sign in using AWS Account credentials](#)

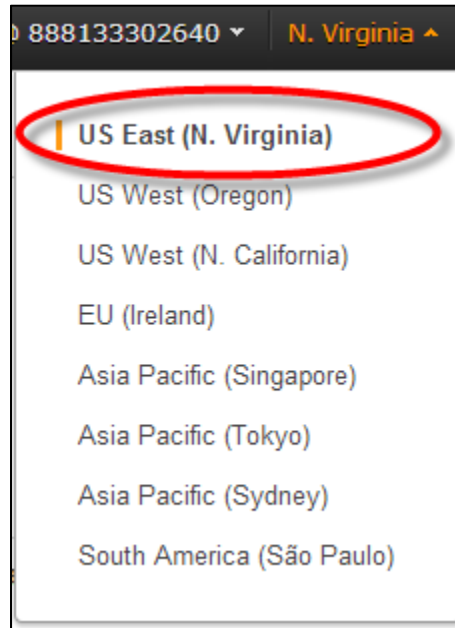
AWS Management Console

8. Select "CloudFormation" from the Console Home.



Confirm your AWS Region

9. Select or confirm that the same AWS Region is already set in the AWS Management Console



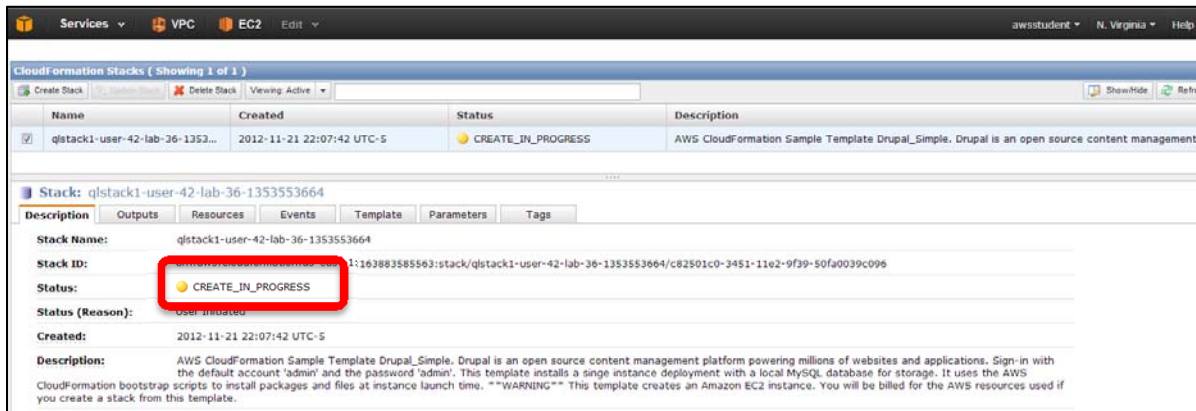
Using Amazon RDS for Applications

We have used a base Amazon Linux build and then used Cloud Formation to bootstrap the instance and your account with all the required tools and configurations required for this lab. You will see your stack status as either `CREATE_IN_PROGRESS` or `CREATE_COMPLETE`. You may want to take 5 minutes to review the different tabs for your stack. You will be able to see the resources that have been created for you, and the template itself.

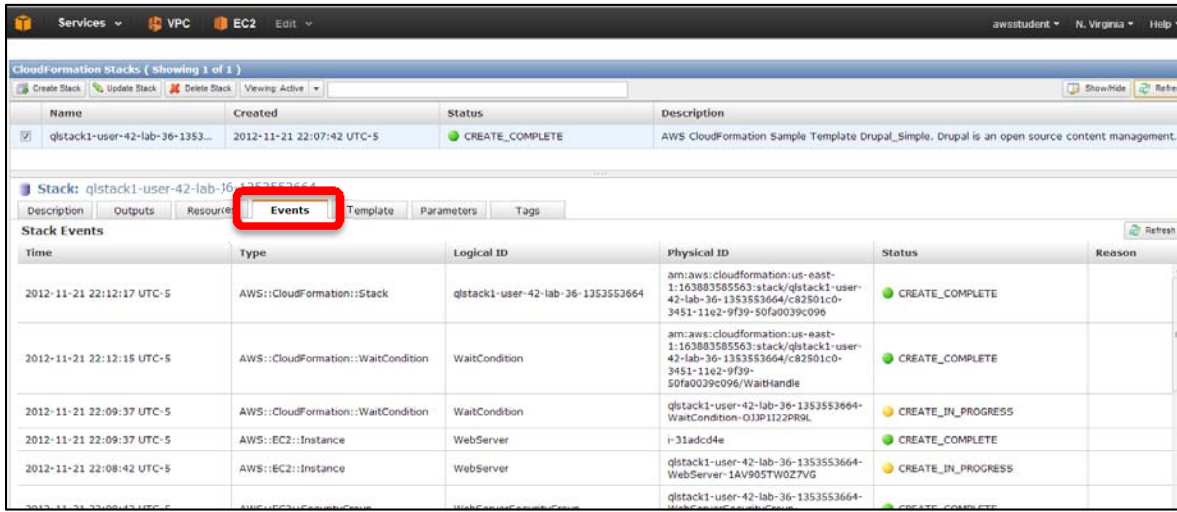
10. Click on the CloudFormation Stack



11. You will see the "Description" of the CloudFormation that is in progress (or has been completed).



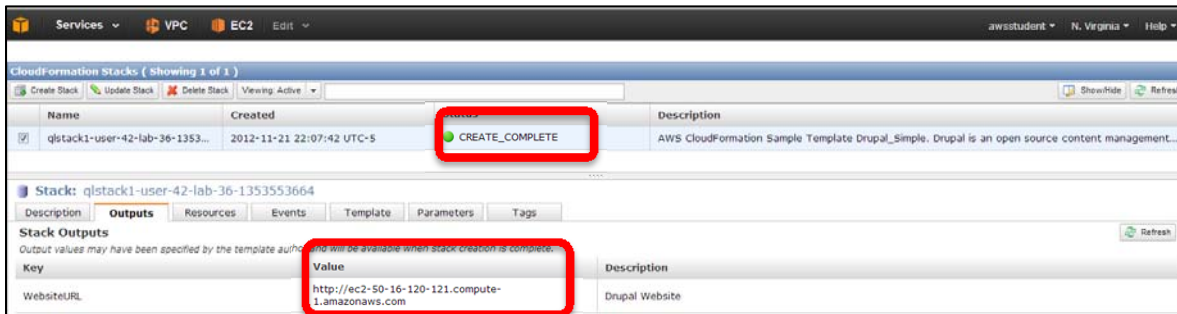
12. Click on the “Events” tab to see Stack Events



Keep refreshing the screen (using Refresh in the upper right corner) until the status of the Stack changes to “CREATE_COMPLETE”

13. Once the status changes to “CREATE_COMPLETE”, click on the “Outputs” tab

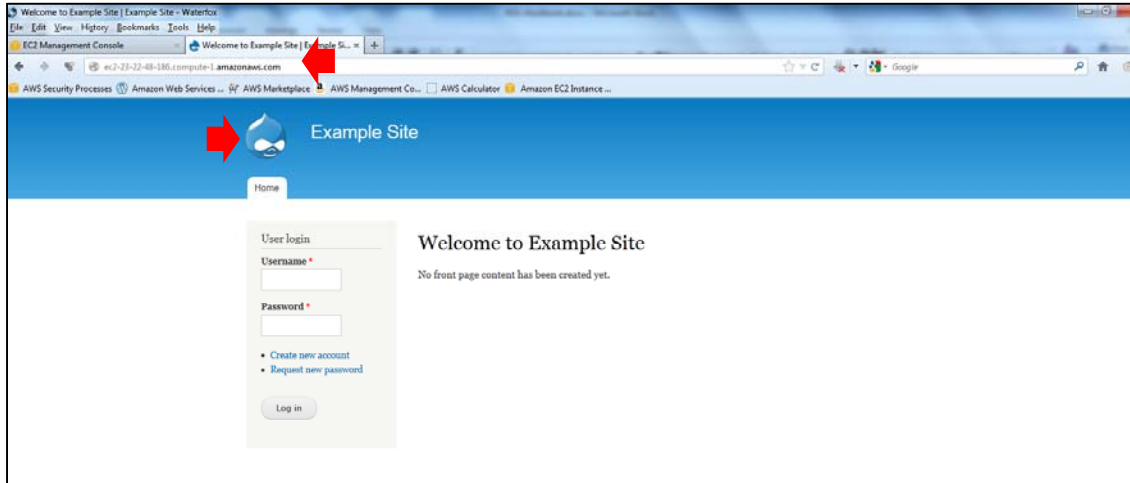
14. Copy the Value of the “WebSiteURL”



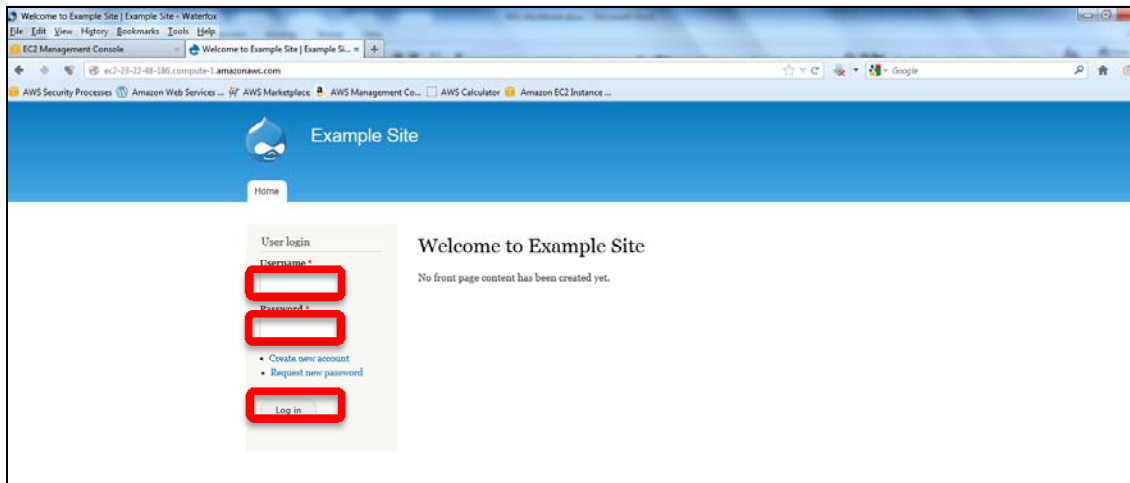
Configure the CMS

The AMI associated to this lab booted as a fully-functional CMS application, so you need to log in and configure a few things. We're not going to discuss how to configure the application, except as it relates to our lab here.

1. Open a new browser window
2. Paste the URL in the browser

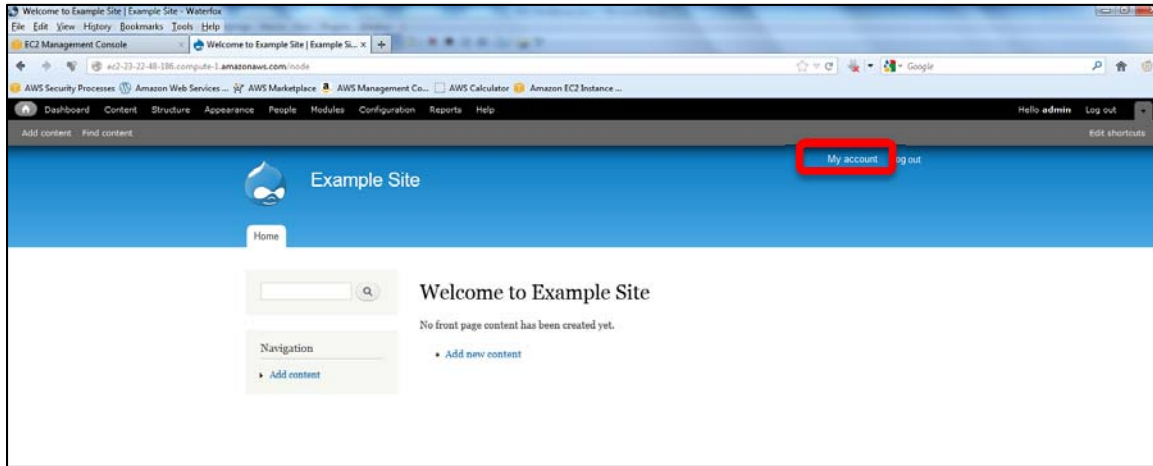


3. Enter the Username "admin" and the Password "admin" then click Log In.

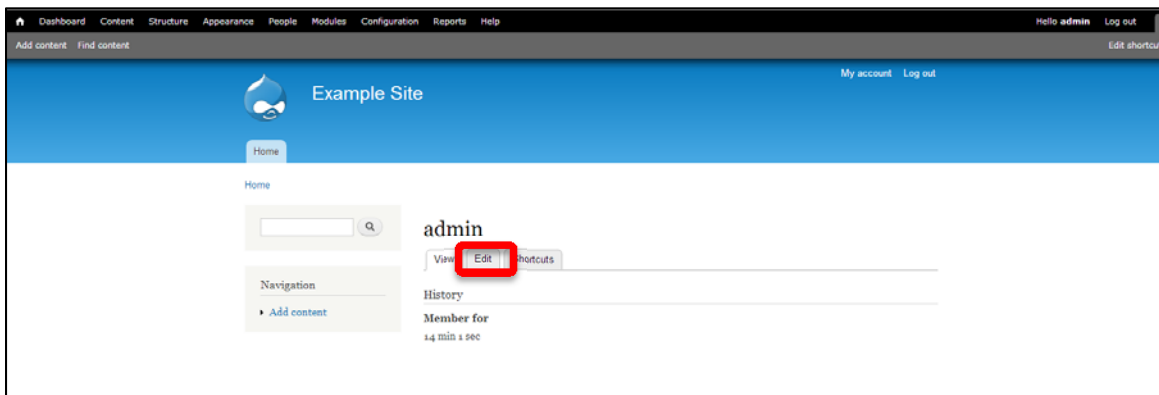


Using Amazon RDS for Applications

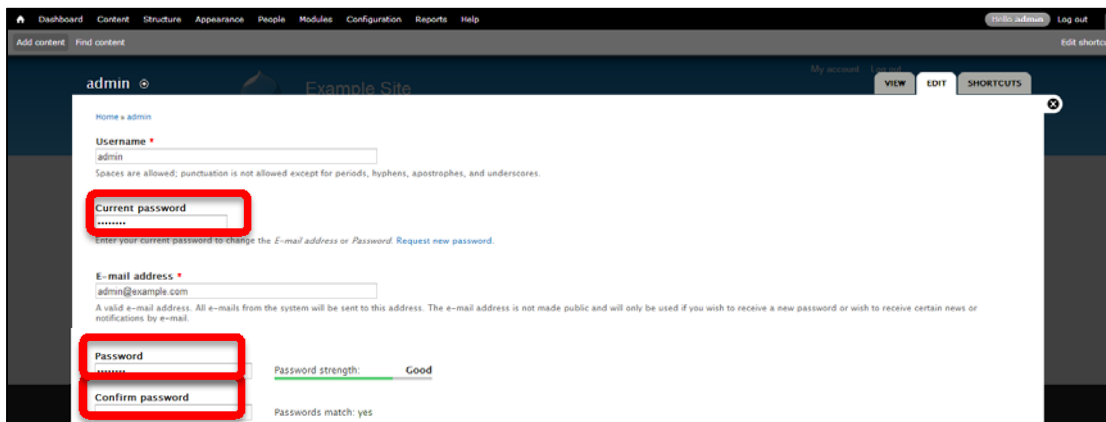
You will end up in the “Example Site” home page



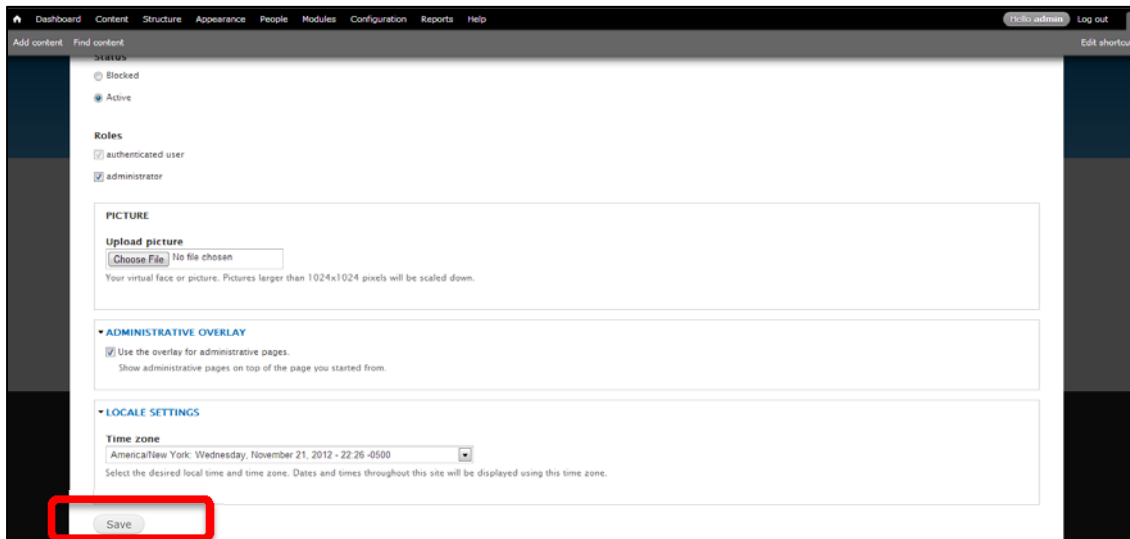
4. Click on “My Account”
5. Click on “Edit”



6. Enter “admin” in the “Current Password” section
7. Enter “admin123” in the “Password” and “Confirm Password” sections

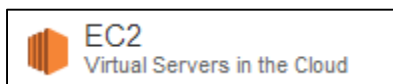


- Click on “Save”



Log into the Drupal CMS instance using SSH

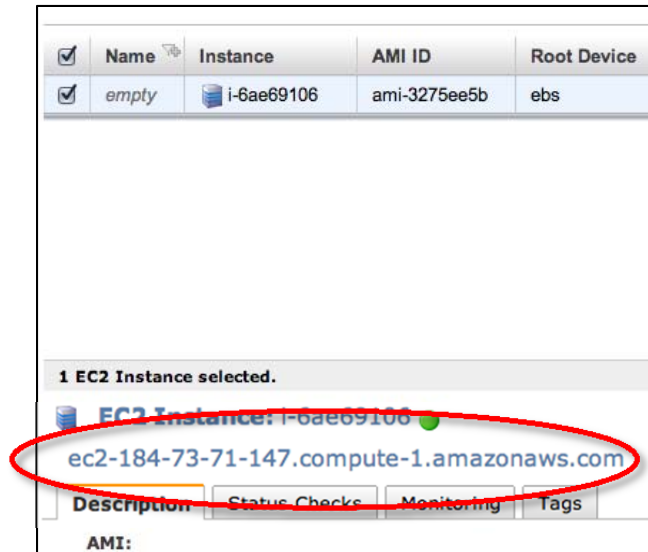
- Go to the EC2 console



- Click on Instances
- Click on the checkbox for the CMS Instance

	Name	Instance	AMI ID	Root Device
<input checked="" type="checkbox"/>	empty	i-6ae69106	ami-3275ee5b	ebs

- Copy the DNS host name for the instance.



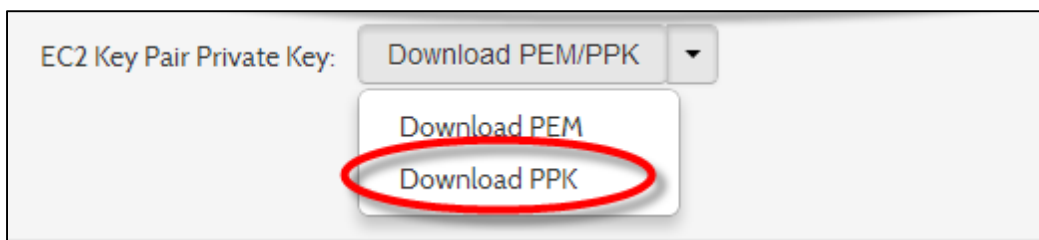
Connect to your EC2 Instance via SSH (Windows)

Download PuTTY

1. Download PuTTY to a location of your choice unless you already have PuTTY.
<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Download your EC2 Key Pair private key file

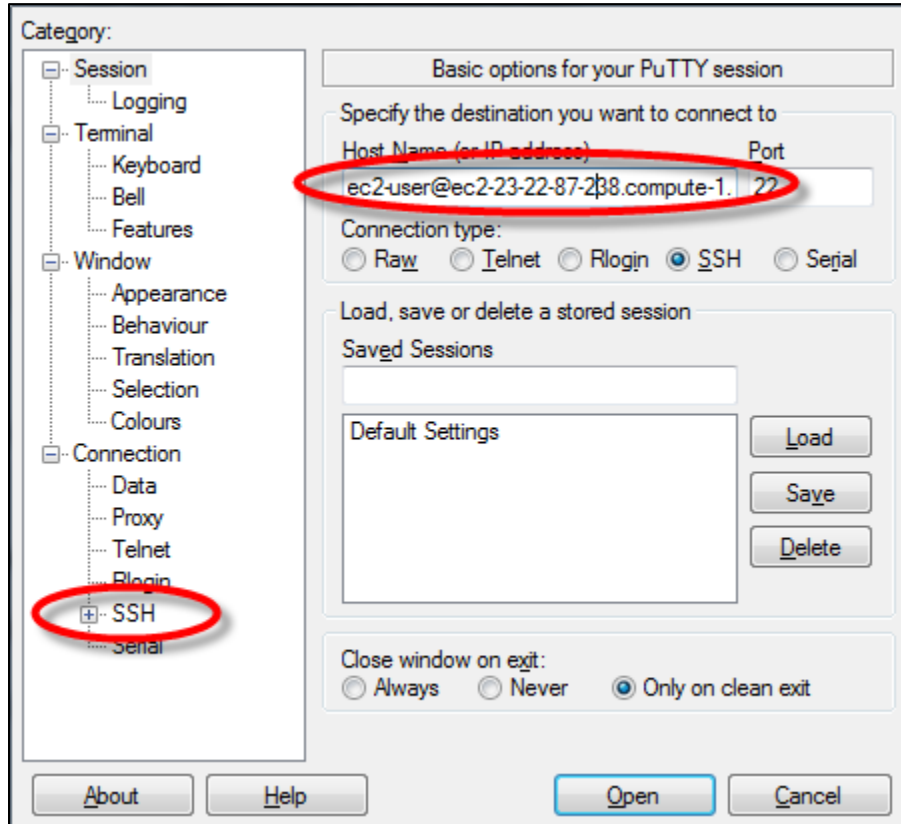
2. Go back to your lab in *qwikLAB*™.
3. Download the *qwikLAB*™ provided EC2 Key Pair private key file in the PuTTY compatible PPK format by clicking on the Download PPK option in the “Download PEM/PPK” drop-down.



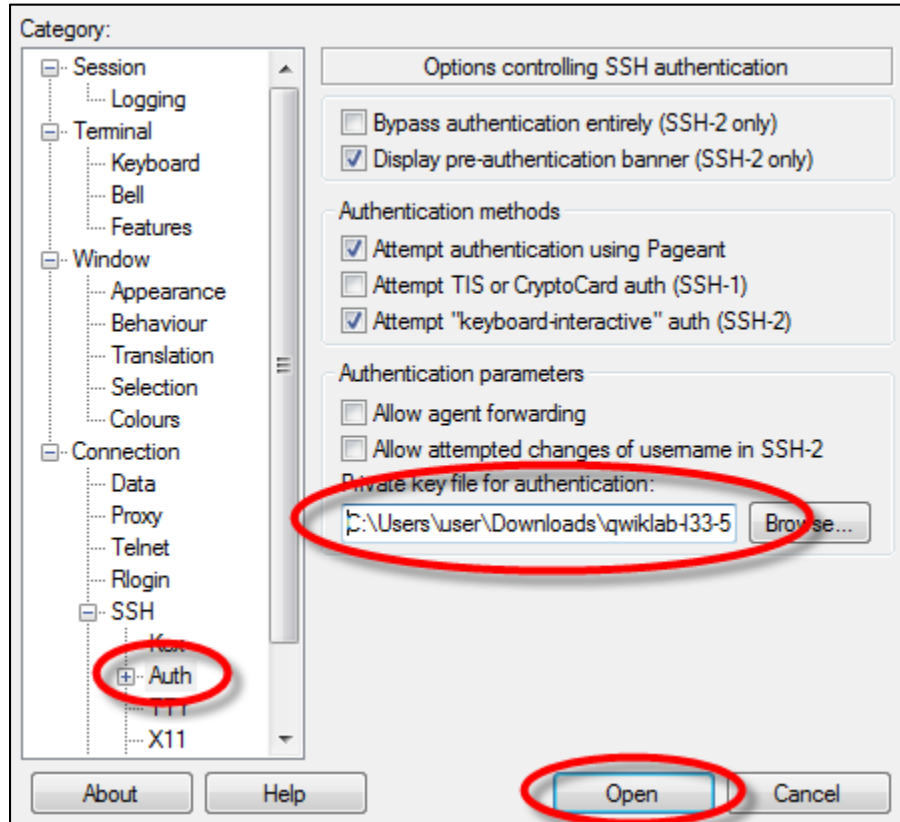
4. Save the file to your Downloads directory (or some other directory of your choice.)

Connect to the EC2 Instance using SSH and PuTTY.

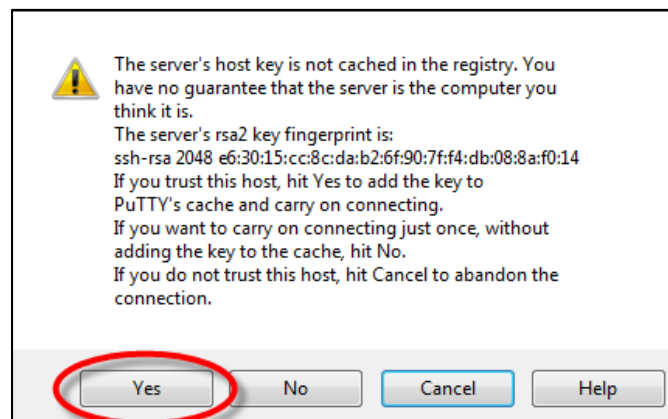
1. Open the putty.exe you downloaded or already had.
2. Enter ec2-user@<your EC2 hostname> into the Host Name input in Putty (Ctrl+v).
3. Expand the SSH category by clicking on it.



4. Select the Auth category by clicking on it (not the + symbol next to it).
5. Click Browse and locate the PPK file (ending in .ppk) in your Downloads directory or whatever other location you chose.
6. Click Open



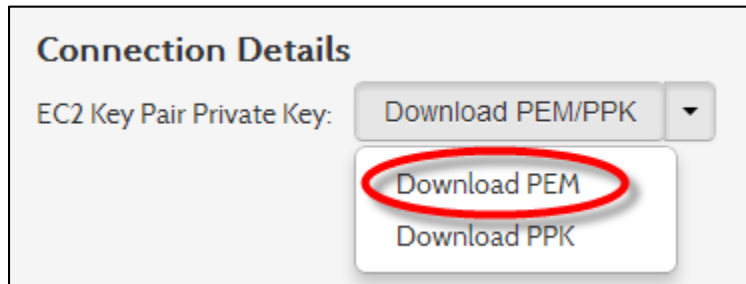
7. Click Yes when prompted to allow a first connection to this remote SSH server.



Connect to your EC2 Instance via SSH (OS X and Linux)

Download your EC2 Key Pair private key file

1. Go back to your lab in *qwikLAB™*.
2. Download the *qwikLAB™* provided EC2 Key Pair private key file in the PEM format by clicking on the Download PEM option in the “Download PEM/PPK” drop-down.



3. Save the file to your Downloads directory (or some other directory of your choice.)

Connect to the EC2 Instance using the OpenSSH CLI client

1. Open the Terminal application.
2. Enter the below commands substituting the path/filename for the .pem file you downloaded from *qwikLAB™* and pasting `ec2-user@<your EC2 hostname>` to substitute the example below.

```
chmod 600 ~/Downloads/qwiklab-l33-5018.pem  
ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@ec2-23-22-87-238.compute-1.amazonaws.com
```

Backup the CMS Database

Before we move on to the next section we should first backup our local database and configuration files.

Backup

1. Backup the Database by typing the following commands:

```
cd ~
```

```
mkdir backup
```

2. Run the following commands to get Drupal MySQL database root password to backup the database.

```
more /var/www/html/sites/default/settings.php | grep drupaldb -A 5 -B 5
```

```
mysqldump -u root -padmin123 drupaldb > backup/backup.sql
```

Note: drupaldb is the name of the database

3. Verify that the backup has completed successfully:

```
cd backup
```

```
ls -l
```

```
tail -f backup.sql
```

You should see Dump completed ... in the screen output

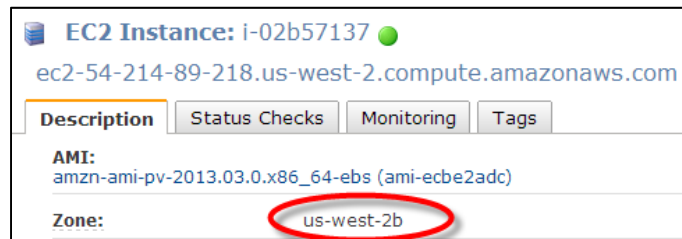
4. Exit out of the command with Ctrl+c

Integrate Drupal CMS Application with Amazon RDS

We have our Drupal CMS system up and running, so let's examine how to make the application scalable and durable. We'll focus on two ways to achieve our goal. In this section we will switch out the database to Amazon Relational Database Service (RDS).

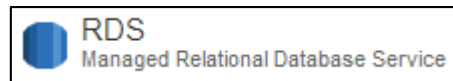
Determine in which Availability Zone your instance is running

1. Go to the management console
2. Click on EC2 Tab
3. Select Instances
4. Select your running CRM instance
5. Select view details in the Description tab
6. Note the Availability Zone (**you will need this later**)



Set Up a Single-AZ Amazon RDS Database

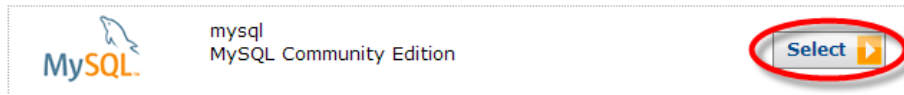
1. Go to the RDS console



2. Click on Launch DB Instance

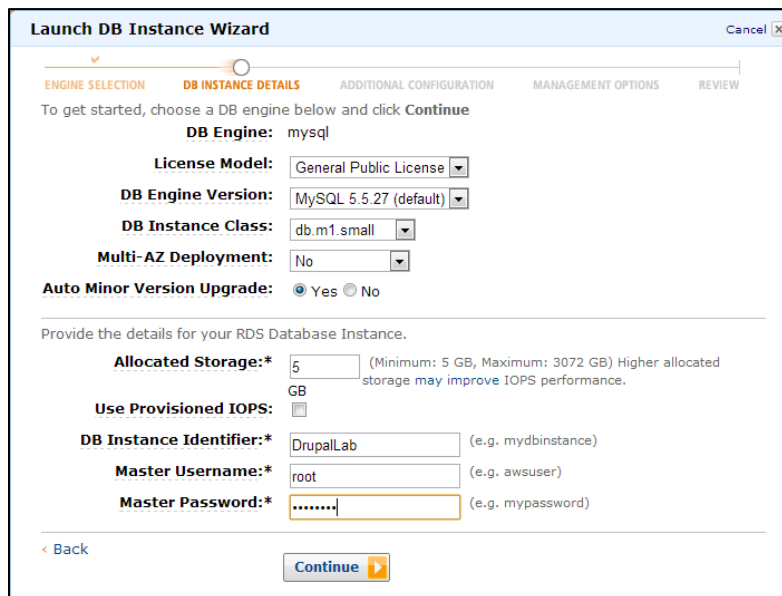


3. Select the MySQL option



4. Configure Wizard Settings

- a. Select General Public License
- b. Select 5.5.27 for engine
- c. Select db.m1.small for instance
- d. Select No for Multi-AZ-Deployment
- e. Select Yes for Auto Minor Version Upgrade
- f. Enter 5GB for Allocated Storage
- g. Use Provisioned IOPS: leave it unchecked
- h. DB Instance Identifier: "DrupalLab"
- i. Master Username: "root"
- j. Master Password: "admin123"



Note: The Master Password must be more than eight characters long.

Using Amazon RDS for Applications

5. Additional Configuration
 - a. Database Name: drupaldb
 - b. Database Port: 3306
 - c. Choose a VPC: Disabled
 - d. Availability Zone: Same as your CMS instance
 - e. Option Group: Disabled
 - f. Parameter Group: Leave the “Default” selection
 - g. Security Group: Leave the “Default” selection
 - h. Click Continue

The screenshot shows the 'Launch DB Instance Wizard' with the 'Additional Configuration' step selected. The wizard has five steps: ENGINE SELECTION, DB INSTANCE DETAILS, ADDITIONAL CONFIGURATION, MANAGEMENT OPTIONS, and REVIEW. The 'ADDITIONAL CONFIGURATION' step is highlighted with a blue bar and a progress indicator. Below the step bar, the text 'Provide the optional additional configuration details below.' is displayed. The configuration fields are as follows: 'Database Name' is 'drupaldb' (with a note '(e.g. mydb)'); 'Database Port' is '3306'; 'Choose a VPC' is 'Not in VPC' (with a note 'Only VPCs with a DB Subnet Group(s) are allowed'); 'Availability Zone' is 'us-west-2b'; 'Option Group' is 'default:mysql-5-5'; 'Parameter Group' is 'default.mysql5.5'; and 'DB Security Group(s)' is 'default'. At the bottom, there are 'Back' and 'Continue' buttons.

Launch DB Instance Wizard Cancel

ENGINE SELECTION DB INSTANCE DETAILS **ADDITIONAL CONFIGURATION** MANAGEMENT OPTIONS REVIEW

Provide the optional additional configuration details below.

Database Name: drupaldb (e.g. mydb)

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

Database Port: 3306

Choose a VPC: Not in VPC Only VPCs with a DB Subnet Group(s) are allowed

Availability Zone: us-west-2b

Option Group: default:mysql-5-5

If you have custom DB Parameter Groups or DB Security Groups you would like to associate with this DB Instance, select them below, otherwise proceed with default settings.

Parameter Group: default.mysql5.5

DB Security Group(s): default

[< Back](#) [Continue >](#)

Using Amazon RDS for Applications

- Leave everything the same on the same screen and click “Continue”

The screenshot shows the 'Launch DB Instance Wizard' window, specifically the 'MANAGEMENT OPTIONS' step. The progress bar at the top indicates the current step. The 'Enabled Automatic Backups' option is set to 'Yes'. Below this, there is a note about backup retention and a link to details. The 'Backup Retention Period' is set to 1 day. The 'Backup Window' and 'Maintenance Window' are both set to 'No Preference'. At the bottom, there are 'Back' and 'Continue' buttons.

Launch DB Instance Wizard [Cancel X]

ENGINE SELECTION DB INSTANCE DETAILS ADDITIONAL CONFIGURATION **MANAGEMENT OPTIONS** REVIEW

Enabled Automatic Backups: ☒ Yes ☐ No

The number of days for which automated backups are retained.

Please note that automated backups are currently **supported for InnoDB storage engine only**. If you are using MyISAM, refer to details [here](#).

Backup Retention Period: 1 days

The daily time range during which automated backups are created if automated backups are enabled

Backup Window: ☐ Select Window ☒ No Preference

The weekly time range (in UTC) during which system maintenance can occur.

Maintenance Window: ☐ Select Window ☒ No Preference

< Back [Continue]

- Review the information and then click on “Launch DB Instance”

The screenshot shows the 'Launch DB Instance Wizard' window, specifically the 'REVIEW' step. It displays a summary of the configuration for a MySQL database instance. The 'Launch DB Instance' button is highlighted. At the bottom, there are 'Back' and 'Launch DB Instance' buttons.

Launch DB Instance Wizard [Cancel X]

ENGINE SELECTION DB INSTANCE DETAILS ADDITIONAL CONFIGURATION MANAGEMENT OPTIONS **REVIEW**

Please review the information below, then click **Launch DB Instance**.

Engine: mysql
Engine Version: MySQL 5.5.27
License Model: general-public-license
Auto Minor Ver. Upgrade: Yes
DB Instance Class: db.m1.small
Multi-AZ Deployment: No
Allocated Storage: 5
Provisioned IOPS: default
DB Instance Identifier: DrupalLab
Master User Name: root
Master User Password: *****

Database Name: drupaldb
Database Port: 3306
Availability Zone: us-west-2b
Option Group: default:mysql-5-5
DB Parameter Group: default.mysql5.5
DB Security Group(s): default
DB Subnet Group:

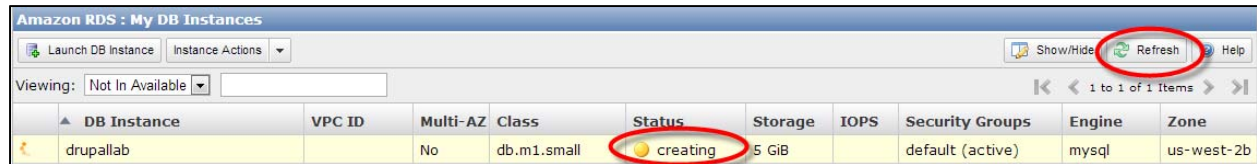
Publicly Accessible: Only applicable with VPC

Backup Retention Period: 1
Backup Window: No Preference
Maintenance Window: No Preference

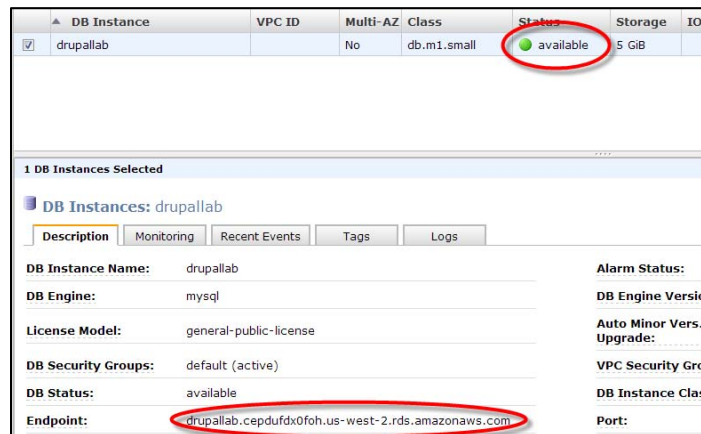
< Back [Launch DB Instance]

8. Click "Close"

Using the DB Instances menu option, wait until the status changes from "Creating" to "Available". Use the Refresh button in the upper right-hand corner. This may take several minutes.



9. Open a plain text editor and copy-paste the database Endpoint URL to it

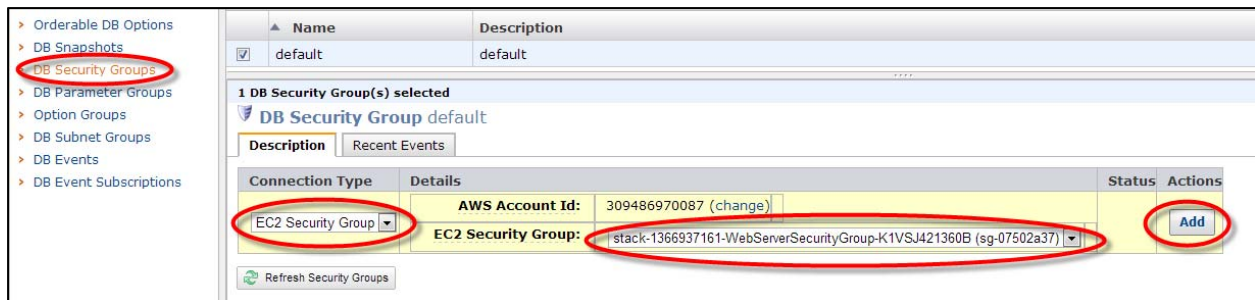


Add Your EC2 Security Group to Database List

The Amazon RDS servers have the same security model as Amazon EC2 overall: trust nothing. Add your Amazon EC2 security group, as follows on the next page. This screen is often confusing to new users.

Note: The name of this security group associated with the Amazon RDS instance is “default”. This “default” security group will be connected to the security group associated with your Amazon EC2 instance.

10. Select DB Security Groups
11. Select the “default” DB Security Group
12. Select EC2 Security Group under Connection Type
13. Select the “qlstack...” EC2 Security Group, not the default group
14. Click Add



Transition the Drupal CMS Instance to use the Amazon RDS database

1. Go back to your SSH session.
See the “Connect to your EC2 Instance ...” sections above if you need directions to connect again.

Import the data from the backup that we performed earlier

2. First we are going to import the data from your existing MySQL database over to Amazon RDS.
3. **Copy** this command into a text file and **change the end point to your RDS instance**

```
mysql -u root -padmin123 --database=drupaldb --host=drupallab.cepdudfx0foh.us-west-2.rds.amazonaws.com < backup/backup.sql
```

The command may not work if you copied and pasted in your SSH window. Try reconstructing the command by typing it into a plain-text editor. The problem is usually extra carriage returns and strange characters instead of spaces.

Reconfiguring the settings.php file to point to the new Amazon RDS database

1. Open the config.php file in vi by typing


```
sudo vi /var/www/html/sites/default/settings.php
```

2. Use the Page Down key to look for a section that looks something as follows:

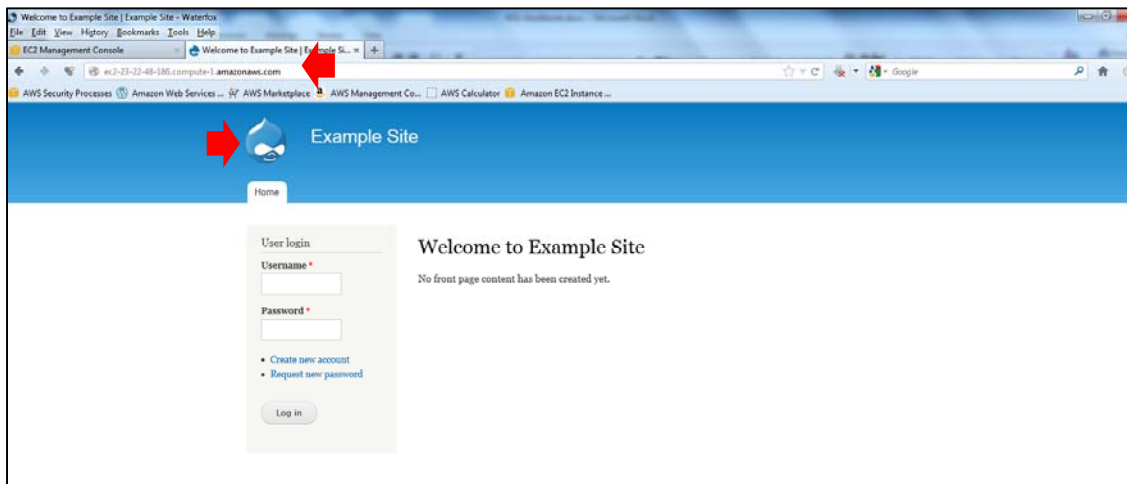
```
$databases = array (  
  'default' =>  
    array (  
      'default' =>  
        array (  
          'database' => 'drupaldb',  
          'username' => 'root',  
          'password' => 'admin123',  
          'host' => 'localhost',  
          'port' => '',  
          'driver' => 'mysql',  
          'prefix' => 'drupal_',  
        ),  
      ),  
    ),  
);
```

3. Edit the 'host' with your Amazon RDS endpoint. Use the endpoint you pasted into your text editor.
 - a. Copy the Amazon RDS endpoint from your text editor.
 - b. Switch back to the SSH window
 - c. Move your cursor to "l" in the "localhost" string; type "cw"; click on "Edit" tab and select "Paste" selection
 - d. Hit "Esc"
 - e. Type :wq! And hit enter
4. The config file is saved with the new Amazon RDS endpoint

```
'host' => 'drupalab.cepdudfx0foh.us-west-2.rds.amazonaws.com',
```

We have now switched the database backend from the local instance to a dedicated and managed database tier. To verify that we are actually reading from the Amazon RDS database, first log into the web frontend of the Drupal CMS instance and click around the user interface. If you get database errors at this point go back through the Amazon RDS steps and verify that you have completed all the steps.

5. Open a browser and paste the Drupal URL in the browser



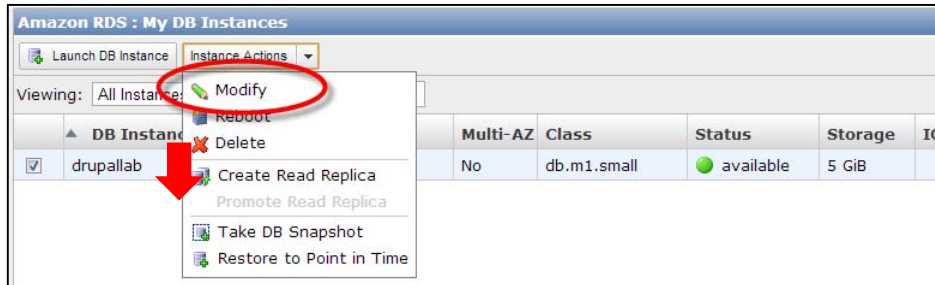
If you do not see this screen you may have configured the Security Group improperly. Check the security group to make sure it allows port 80 for HTTP connections.

If the web front end is working properly you will not notice a change from when the database was on the same machine as the web frontend. To verify that you are indeed reading from the Amazon RDS instance go to the security groups section in the Amazon RDS tab and revoke the "%qlstack..." security group connection you made previously. Now when you go back to the web frontend you will not be able to login and it will throw DB errors. Authorize the "%qlstack..." again and everything should work properly.

Change the Amazon RDS Instance Size

Scaling up and down with Amazon RDS is simple via the AWS Console. You can grow the database or change the underlying server size, etc. – all from the AWS Console.

1. Go to the management console
2. Click on RDS Tab
3. Select DB Instances
4. Select your RDS instance
5. Click on Modify



6. Change The DB instance Class to db.m1.large
7. Select yes to make the DB a Multi-AZ Deployment
8. Check “Apply Immediately”
9. Click Continue.

10. Click Modify DB Instance.

DB Instance	VPC ID	Multi-AZ	Class	Status	Storage
drupallab		No	db.m1.small	modifying	5 GiB

Note: This will take a few minutes to modify

What if the entire availability zone becomes unavailable? A rare occurrence, but certainly possible. By selecting Yes for the Multi-AZ Deployment option we have made the database highly available.

At this point you have 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 primary instance for IO. Should the primary fail, AWS will automatically promote the secondary server to primary while simultaneously recovering the errant server. Your connection string will remain the same, so other than a hiccup there is no effect on your application.

Conclusion

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.

You have started to use the AWS Management Console or Amazon RDS APIs to launch a Database Instance (**DB Instance**), selecting the DB Engine (MySQL, Oracle or SQL Server), License Type, DB Instance class and storage capacity that best meets your needs.

You also have connected to your DB Instance using your favorite database tool or programming language.

You have also learnt how you are able to scale the compute resources or storage capacity associated with your Database Instance (DB Instance) via a single API call.

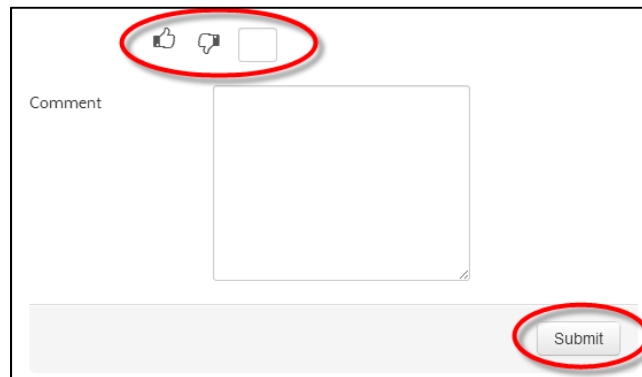
Finally, we you will learn how to have a very durable database, and you can even perform functions such as backups without incurring downtime.

End Lab

1. Sign-out of the AWS Management Console.
2. Click the End Lab button in *qwikLAB™*.



3. Give the lab a thumbs-up/down, or enter a comment and click Submit

A feedback form interface. At the top, there are three icons: a thumbs-up, a thumbs-down, and a square box, all enclosed in a red oval. Below these icons is a text input field labeled "Comment". To the right of the "Comment" label is a large, empty rectangular text area. At the bottom right of the form, there is a "Submit" button, which is also enclosed in a red oval.

You may report any errors in this lab to aws-course-feedback@amazon.com.