

# LAB 02: APP SCALING ON AMAZON WEB SERVICES

Auteurs : Jeremy Zerbib, Samuel Mettler

Date : March 19, 2020

## TASK 1: CREATE A DATABASE USING THE RELATIONAL DATABASE SERVICE (RDS)

1. In the EC2 console create a *Security Group* with a name of the form *yourlastname-Drupal-DB-us-east1* and open the TCP port on 3306 (MySQL default port).

Create security group [Info](#)

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

**Basic details**

Security group name [Info](#)  
zerbib-Drupal-DB-us-east1  
Name cannot be edited after creation.

Description [Info](#)  
Security group for CLD lab2

VPC [Info](#)  
vpc-d1b380ab

**Inbound rules** [Info](#)

Type <a href="#">Info</a>	Protocol <a href="#">Info</a>	Port range <a href="#">Info</a>	Source <a href="#">Info</a>	Description - optional <a href="#">Info</a>
Custom TCP	TCP	3306	Custom	

[Add rule](#) [Delete](#)

2. Open the RDS console. Make sure to create the database in the same region as the Drupal master instance, switch the console to the region in which you created the Drupal master instance if it is not done.
3. Launch a DB instance: Click **Create database** and provide the following answers (leave any field not mentioned at its default value):
  - Choose a database creation method
    - Select **Standard Create**
  - Engine options
    - Engine type: **MySQL**
    - Edition: **MySQL Community**
    - Version: **MySQL 5.7.22**

## Create database

### Choose a database creation method [Info](#)

☒ **Standard Create**

You set all of the configuration options, including ones for availability, security, backups, and maintenance.

☐ **Easy Create**

Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

### Engine options

#### Engine type [Info](#)

☐ Amazon Aurora



☒ **MySQL**



☐ MariaDB



☐ PostgreSQL



☐ Oracle



☐ Microsoft SQL Server



#### Edition

☒ **MySQL Community**

#### Version [Info](#)

MySQL 5.7.22

- Templates
  - Select **Dev/Test - MySQL**
- Settings
  - DB Instance Identifier: **yourlastname-Drupal**
  - Master username: **admin**
  - Master Password: Invent a password and write it down (**lab2-cld2020**)

## Templates

Choose a sample template to meet your use case.

☐ **Production**  
Use defaults for high availability and fast, consistent performance.

☒ **Dev/Test**  
This instance is intended for development use outside of a production environment.

☐ **Free tier**  
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

## Settings

### DB Instance Identifier [Info](#)

Type a name for your DB instance. The name must be unique cross all DB instances owned by your AWS account in the current AWS Region.

zerbib-Drupaé

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens (1 to 15 for SQL Server). First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

### ▼ Credentials Settings

#### Master username [Info](#)

Type a login ID for the master user of your DB instance.

admin

1 to 16 alphanumeric characters. First character must be a letter

☐ **Auto generate a password**

Amazon RDS can generate a password for you, or you can specify your own password

#### Master password [Info](#)

lab2-cld2020

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), "(double quote) and @ (at sign).

#### Confirm password [Info](#)

lab2-cld2020

**There's been a typo in the Identifier field in the screenshot above but it has been corrected in the lab**

- DB instance size
  - DB instance class
- Select **Burstable classes**
- Select **db.t2.micro**
  - Storage
    - Storage type: **General Purpose (SSD)**
    - Allocated Storage: **20 GB**
  - Storage Autoscaling:
    - Enable storage autoscaling: **unchecked**
  - Availability & durability
    - Multi-AZ deployment
      - **Do not create a standby instance**

### DB instance size

**DB instance class** [Info](#)  
Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

☐ Standard classes (includes m classes)  
☐ Memory Optimized classes (includes r and x classes)  
☒ Burstable classes (includes t classes)

db.t2.micro  
 1 vCPUs   1 GiB RAM   Not EBS Optimized

☐ Include previous generation classes

---

### Storage

**Storage type** [Info](#)  
General Purpose (SSD)

**Allocated storage**  
20 GiB

(Minimum: 20 GiB, Maximum: 16384 GiB) Higher allocated storage [may improve](#) IOPS performance.

ⓘ Provisioning less than 100 GiB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. [Learn more](#)

**Storage autoscaling** [Info](#)  
Provides dynamic scaling support for your database's storage based on your application's needs.

☐ **Enable storage autoscaling**  
Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

---

### Availability & durability

**Multi-AZ deployment** [Info](#)

☐ Create a standby instance (recommended for production usage)  
 Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.  
☒ Do not create a standby instance

- Connectivity
  - Virtual Private Cloud (VPC)
    - Choose **Default VPC**
    - Additional connectivity configuration
      - Subnet group : **default**
      - Publicly accessible : **Yes**
      - VPC security group
        - Select **Choose existing**
    - Remove **default**
  - Add **yourname-Drupal-DB**
  - Leave the rest as it is

## Connectivity

### Virtual Private Cloud (VPC) [Info](#)

VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-173a096d) ▼

Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change the VPC selection.

### ▼ Additional connectivity configuration

### Subnet group [Info](#)

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default ▼

### Publicly accessible [Info](#)

☒ **Yes**  
 Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☐ **No**  
 RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

### VPC security group

Choose one or more RDS security groups to allow access to your database. Ensure that the security group rules allow incoming traffic from EC2 instances and devices outside your VPC. (Security groups are required for publicly accessible databases.)

☒ **Choose existing**  
 Choose existing VPC security groups

☐ **Create new**  
 Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

zerbib-Drupal-DB-us-east1 ✕

### Availability zone [Info](#)

No preference ▼

### Database port [Info](#)

TCP/IP port the database will use for application connections.

3306

- Database authentication
- Database authentication options
- Check **Password authentication**
  - Write down the estimated monthly costs that are displayed.

### Database authentication

Database authentication options [Info](#)

☒ Password authentication  
Authenticates using database passwords.

☐ Password and IAM database authentication  
Authenticates using the database password and user credentials through AWS IAM users and roles.

► **Additional configuration**

Database options, backup enabled, backtrack disabled, Enhanced Monitoring enabled, maintenance, CloudWatch Logs, delete protection disabled

### Estimated monthly costs

DB Instance	12.41 USD
Storage	2.30 USD
<b>Total</b>	<b>14.71 USD</b>

This billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, IOs (if applicable), or data transfer.

Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#).

**The estimated monthly cost is 14.71\$.**

After launching the DB instance return to the instances view and wait for the DB instance to be created.

- In the RDS console select the newly created DB instance and write down the **Endpoint** address.

The Endpoint address is : **zerbib-drupal.chkfvjvvdjw1.us-east-1.rds.amazonaws.com**

- Test whether the database can be reached from the Drupal master instance.
  - Log into the Drupal master instance.
- Using the database's endpoint address (**without** the port number) and the master password you wrote down run the command:

```
mysql --host=endpoint_address --user=<rds_master_username> --password=<rds_master_password>
```

You should see a welcome message and the MySQL command line prompt `mysql>`. Type `quit` to exit.

```
~ > ~/B/H/S/CLD/Labos/CLD_Labo2 > on P master !1 ?1 ssh -i "mint-key-pair-NV.pem" ubuntu@ec2-3-86-228-8.compute-1.amazonaws.com

The authenticity of host 'ec2-3-86-228-8.compute-1.amazonaws.com (3.86.228.8)' can't be established.
ECDSA key fingerprint is SHA256:VQUMA+wm+/GSj9mhNVizWKYmaz3zPbBqB/gAV6dj/A4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-86-228-8.compute-1.amazonaws.com,3.86.228.8' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1060-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Mar 11 13:28:49 UTC 2020

System load: 0.39          Processes:           99
Usage of /:  29.9% of 7.69GB Users logged in:       0
Memory usage: 33%        IP address for eth0: 172.31.81.95
Swap usage:  0%

7 packages can be updated.
0 updates are security updates.

Last login: Thu Feb 27 11:39:07 2020 from 193.134.219.71
ubuntu@ip-172-31-81-95:~$ mysql --host=zerbib-drupal.chkfvjvvdjw1.us-east-1.rds.amazonaws.com --user=admin --password=lab2-cld2020
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.7.22-log Source distribution

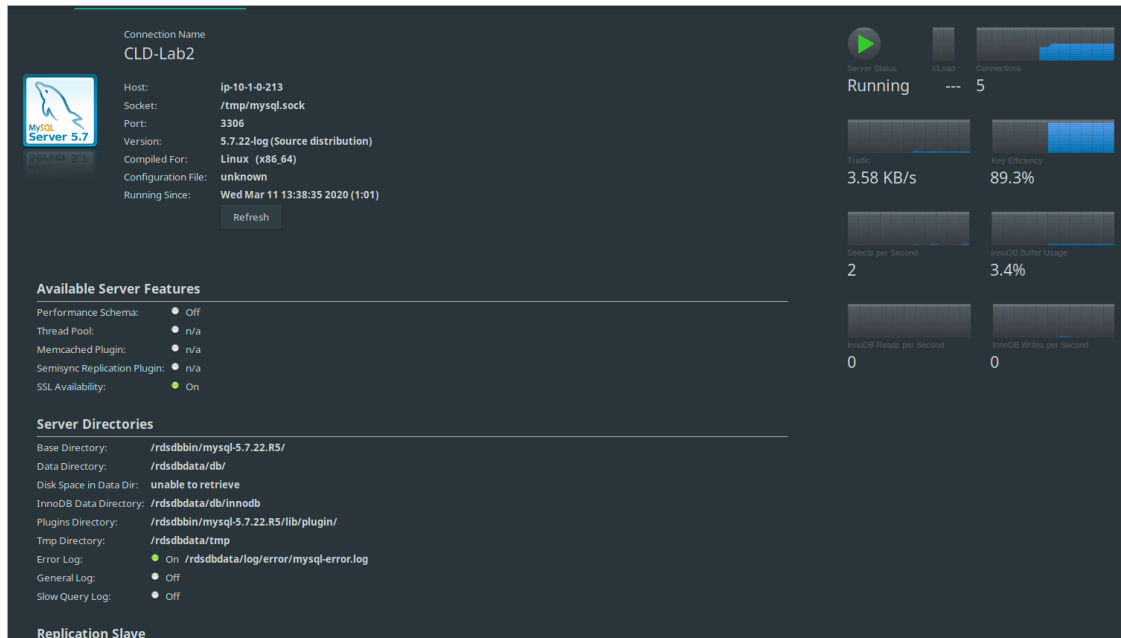
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> quit
Bye
ubuntu@ip-172-31-81-95:~$
```

6. **Optional:** On your local machine download and install the **MySQLWorkbench** administration tool from <http://www.mysql.com/products/workbench/> and use it to connect to the database.



## Livrables

- Copy the estimated costs you calculated and add it to the report

The estimated monthly costs are up to 14.71\$.

- Compare the costs of your RDS instance to a continuously running EC2 instance of the same size using the AWS calculator. (Don't forget to uncheck the Free Usage Tier checkbox at the top.)

FREE USAGE TIER: New Customers get free usage tier for first 12 months

**Services** Estimate of your Monthly Bill (\$ 6.25)

Choose region: US East (N. Virginia) Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances. [Clear Form](#)

**Compute: Amazon EC2 Instances:**

Description	Instances	Usage	Type	Billing Option	Monthly Cost
	1	100 % Utilized/M	Linux on t2.nano	On-Demand (No Co)	\$ 4.25

[Add New Row](#)

**Compute: Amazon EC2 Dedicated Hosts:**

Description	Number of Hosts	Usage	Type	Billing Option

[Add New Row](#)

**Storage: Amazon EBS Volumes:**

Description	Volumes	Volume Type	Storage	IOPS	Baseline Throughput	Snapshot Storage
	1	General Purpose SSD (gp2)	20 GB	100	128 MB/sec	0 GB-month of Storage

[Add New Row](#)

**Compute: Amazon Elastic Graphics:**

Description	Number of Elastic Graphics	Usage	Elastic Graphics Size and Memory

[Add New Row](#)

**Additional T2/T3 Unlimited vCPU Hours per month:**

For Linux, RHEL and SLES: 0

For Windows and Windows with SQL Web: 0

**Elastic IP:**

☒ Enter values below ☐ Calculate

Total time the additional Elastic IPs are attached to running EC2 instances\*: 0 Hours/Month

Total Non-attached time for all the Elastic IP: 0 Hours/Month

Number of Elastic IP Remaps: 0 Per Month

We can see that for an *EC2 instance*, it will cost roughly 6.25\$ for a monthly usage at 100%.

FREE USAGE TIER: New Customers get free usage tier for first 12 months

**Services** Estimate of your Monthly Bill (\$ 35.74)

Choose region: US East (N. Virginia) Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. [Clear Form](#)

**Amazon RDS On-Demand DB Instances:**

Description	DB Instances	Usage	DB Engine and License	Class and Deployment	Storage	I/O	Backtrack
	1	100 % Utilized/M	MySQL	db.t2.micro	General P.	Provisioned IOPS: 0	

[Add New Row](#)

**Amazon RDS Aurora Global Database Secondary Region Instances:**

Description	DB Instances	Usage	DB Engine and License	Class	Storage	I/O

[Add New Row](#)

\* Note: The above table is to estimate secondary region for Amazon RDS Aurora global Database. Estimate your primary region in the respective region under Amazon RDS On-Demand DB Instances or under Amazon RDS Reserved Instances.

**Amazon RDS Aurora Serverless:**

Description	Usage	DB Engine	Number of Aurora Capacity Unit(ACU)	Storage	I/O

[Add New Row](#)

**Additional Backup Storage (Free backup storage up to 100% of provisioned Storage):**

Backup Type	Backup Storage

[Add New Row](#)

**Amazon RDS Reserved DB Instances:**

Description	DB Instances	DB Engine and License	Class and Deployment	Offering and Term	Storage	I/O	Backtrack

[Add New Row](#)

**Data Transfer:**

Inter-Region Data Transfer Out: 0 GB/Month

Data Transfer Out: 0 GB/Month

Data Transfer In: 0 GB/Month

We can see that the monthly costs for a *RDS instance* will costs roughly 35.74\$ running at 100%.

- In a two-tier architecture the web application and the database are kept separate and run on different hosts. Imagine that for the second tier instead of using RDS to store the data you would create a virtual machine in EC2 and install and run yourself a database on it. If you were the Head of IT of a medium-size business, how would you argue in favor of using a database as a service instead of running your own database on an EC2 instance? How would you argue against it?

Running a *RDS instance* as a service instead of running our own database could be the best idea for the company as we need availability for our clients. If one instance is down, it does not mean that the other is also down.

On the other hand, on a cost basis, it would be wiser to run our own database on an EC2 instance. It is way cheaper to run an EC2 instance as stated above.

- Copy the endpoint address of the database into the report.

The endpoint name is : **zerbib-drupal.chkfvjvvdjw1.us-east-1.rds.amazonaws.com**

## TASK 2: CONFIGURE THE DRUPAL MASTER INSTANCE TO USE THE RDS DATABASE

In this task you will migrate the content of the local MySQL database to the newly created RDS database and change Drupal's database configuration to use the RDS database.



## CHANGE DRUPAL'S DATABASE CONFIGURATION

1. Log into the Drupal master instance.
2. Stop the web server by typing:

```
sudo systemctl stop apache2
```

3. To change Drupal's configuration parameters to point to the RDS database you will need to change the current configuration. At the end of the file **settings.php** found in the drupal folder (`/var/www/html/drupal/sites/default/`), you will find the current lines :

```
$databases['default']['default'] = array (  
  'database' => '<database_name>',  
  'username' => '<username>',  
  'password' => '<password>',  
  'prefix' => '',  
  'host' => 'localhost',  
  'port' => '3306',  
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',  
  'driver' => 'mysql',  
);
```

Change it to :

```
$databases['default']['default'] = array (  
  'database' => '<rds_db_name>',  
  'username' => '<rds_master_username>',  
  'password' => '<rds_master_password>',  
  'prefix' => '',  
  'host' => '<ip_address_of_rds_database>',  
  'port' => '3306',  
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',  
  'driver' => 'mysql',  
);
```

```
$databases['default']['default'] = array (  
  'database' => 'zerbib-drupal',  
  'username' => 'admin',  
  'password' => 'lab2-cld2020',  
  'prefix' => '',  
  'host' => '3.86.228.8',  
  'port' => '3306',  
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',  
  'driver' => 'mysql',  
);
```

4. To make the authentication in the RDS database less strict connect to the database, create an additional user `'<rds_username>@'%'` and give this user access rights to the `<rds_db_name>` database.

Perform the following steps:

- Launch the `mysql` command to connect to the RDS database (it's the same command as the verification step after creating the RDS database):

```
mysql --host=endpoint_address --user=<rds_master_username> --password=
<rds_master_password>
```

You should see a welcome message and the MySQL command line prompt `mysql>`.

5. On the `mysql>` command prompt run the following three commands where `<rds_password>` is the database password for user `<rds_username>` you wrote down earlier:

```
CREATE USER '<rds_username>'@'%' IDENTIFIED BY '<rds_password>';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE
TEMPORARY TABLES, LOCK TABLES ON <rds_db_name>.* TO '<rds_username>'@'%'
IDENTIFIED BY '<rds_password>';
FLUSH PRIVILEGES;
```

```
ubuntu@ip-172-31-81-95:~$ mysql --host=endpoint_address --user=<rds_master_username> --password=<rds_master_password>
-bash: syntax error near unexpected token 'newline'
ubuntu@ip-172-31-81-95:~$ mysql --host=zerbib-drupal.chkfvjvvdjw1.us-east-1.rds.amazonaws.com --user=admin --password=lab2-cld2020
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7109
Server version: 5.7.22-log Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE zerbib-drupal
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '-drupal' at line 1
mysql> CREATE DATABASE zerbib_drupal
-> ;
Query OK, 1 row affected (0.02 sec)

mysql> CREATE USER 'test'@'%' IDENTIFIED BY 'test';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test'@'%'
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK TABLES ON zerbib_drupal.* TO 'test'@'%' IDENTIFIED BY 'test';
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

6. Disconnect from the RDS database by typing `\quit` and verify that user `<rds_username>` can connect to the database by typing

```
mysql --host=endpoint_address --user=<rds_username> --password=<rds_password>
```

```
ubuntu@ip-172-31-81-95:~$ mysql --host=zerbib-drupal.chkfvjvvdjw1.us-east-1.rds.amazonaws.com --user=test --password=test
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7111
Server version: 5.7.22-log Source distribution

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
bye
ubuntu@ip-172-31-81-95:~$
```

## Migrate the database content to RDS

To migrate the data currently stored in the MySQL database of the Drupal master instance into the RDS database, perform the following steps.

1. Log into the Drupal master instance.
2. Type the following command to migrate the database content from the local MySQL database to the RDS database.

```
mysql -uadmin -p$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 32 | tr -d '\n' | xargs echo) zerbib-drupal.chkfvjvvdjw1.us-east-1.rds.amazonaws.com --user=test --password=test zerbib_drupal
mysql: [Warning] Using a password on the command line interface can be insecure.mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
mysqldump --add-drop-table --user=<localhost_db_username> --password=
<localhost_db_password> <localhost_database_name> | mysql --
host=endpoint_address --user=<rds_username> --password=<rds_password>
<rds_db_name>
```

The command should complete without errors.

3. Start the web server by typing:

```
sudo systemctl start apache2
```

4. Verify the database configuration by navigating with your browser to the Drupal home page at `http://hostname/<database_name>/`.

## DELIVERABLE 2

- Copy the part of `settings.php` that configures the database into the report.

```
$databases['default']['default'] = array (
  'database' => 'zerbib_drupal',
  'username' => 'admin',
  'password' => 'lab2-cld2020',
  'prefix' => '',
  'host' => '3.84.43.137',
  'port' => '3306',
  'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',
  'driver' => 'mysql',
);
```

## TASK 3: CREATE A CUSTOM VIRTUAL MACHINE IMAGE

Now that you have properly configured the Drupal master instance, you will save it into a virtual machine image. This image will be used later to create new instances with the exact same configuration.

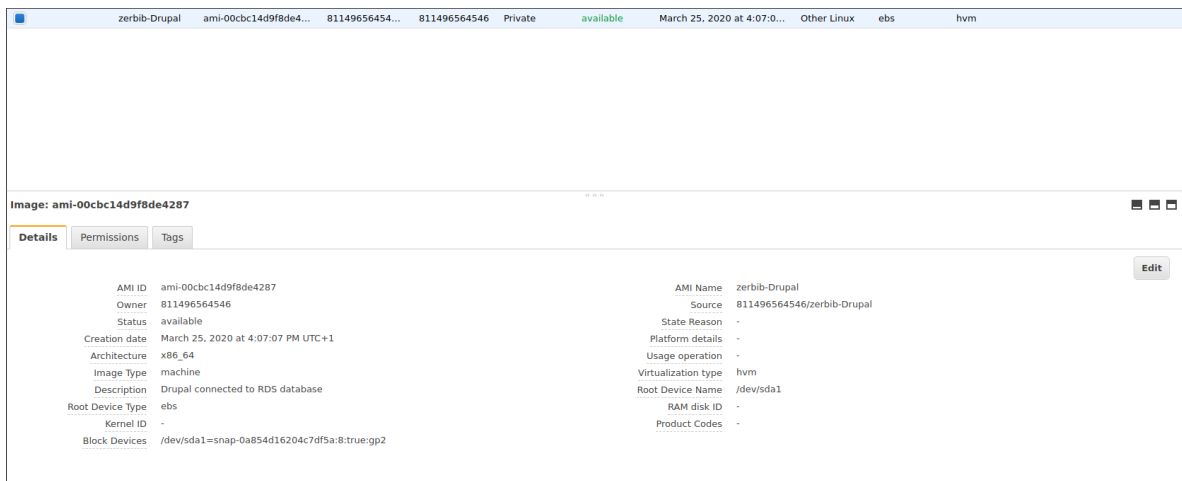
1. In the EC2 console bring up the **Instances** panel and select the Drupal master instance.
2. Bring up the context menu and select **Image > Create Image**. Provide the following answers (leave any field not mentioned at its default value):
  - Image Name: **yourlastname-Drupal**.
  - Image Description: **Drupal connected to RDS database**.

Click **Create Image**. The instance will shut down temporarily and the image will be created.

3. In the console bring up the **AMIs** panel. Wait until the status of the AMI goes from **pending** to **available**.

## DELIVERABLE 3:

- Copy a screenshot of the AWS console showing the AMI parameters into the report.



## TASK 4: CREATE A LOAD BALANCER

In this task you will create a load balancer in AWS that will receive the HTTP requests from clients and forward them to the Drupal instances.

1. Verify and note in which availability zone your instance is running.
2. In the EC2 console bring up the **Load Balancers** panel. Click on **Create Load Balancer**. Provide the following answers (leave any field not mentioned at its default value):
  - Choose : **Application Load Balancer**
  - Configure Load Balancer
    - Name: **yourlastname-loadBalancer**.
    - Scheme: **internet-facing**
    - IP address type: **ipv4**
    - Listeners
      - Make sure the protocol **HTTP** on port **80** is set, else set it.

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name

Scheme ☒ internet-facing  
☐ internal

IP address type

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
<input type="text" value="HTTP"/>	<input type="text" value="80"/>

Availability Zones

- Availability Zones
  - Select 2 availability zones. Make sure that one of them corresponds to where your instance is running.

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC

Availability Zones

<input checked="" type="checkbox"/> us-east-1a	<input type="text" value="subnet-4750b266"/>	<input type="text" value="IPV4 address"/>	Assigned by AWS
<input checked="" type="checkbox"/> us-east-1b	<input type="text" value="subnet-15229598"/>	<input type="text" value="IPV4 address"/>	Assigned by AWS
<input type="checkbox"/> us-east-1c	<input type="text" value="subnet-4468881b"/>		
<input checked="" type="checkbox"/> us-east-1d	<input type="text" value="subnet-31f13547"/>	<input type="text" value="IPV4 address"/>	Assigned by AWS
<input type="checkbox"/> us-east-1e	<input type="text" value="subnet-79d7bc45"/>		
<input type="checkbox"/> us-east-1f	<input type="text" value="subnet-c507d6cb"/>		

- Configure Security Groups
  - Use the same security group as the Drupal instance.

Assign a security group:

☐ Create a new security group

☒ Select an existing security group

Filter VPC security groups

Security Group ID	Name	Description	Actions
<input type="checkbox"/> sg-0df3e726	default	default VPC security group	<a href="#">Copy to new</a>
<input type="checkbox"/> sg-0c71397b56c093eda	launch-wizard-1	launch-wizard-1 created 2020-03-11T13:54:10.479+01:00	<a href="#">Copy to new</a>
<input checked="" type="checkbox"/> sg-056648afe863e321	Security_Group_NV_1	default security group for web server	<a href="#">Copy to new</a>
<input type="checkbox"/> sg-01f1540bd0e6d361e	zerbib-Drupal-DB-us-east1	Security group for CLD Lab 2	<a href="#">Copy to new</a>

- Configure Routing
  - Target Group : **New target group**
  - Name : **yourLastName-target-group**
  - Target Type: Instance
  - Advanced Health check settings
    - Healthy threshold : **2**
    - Interval : **10** seconds

## Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol

### Target group

<b>Target group</b> ⓘ	New target group ▼
<b>Name</b> ⓘ	zerbib-target-group
<b>Target type</b>	<input checked="" type="radio"/> Instance <input type="radio"/> IP <input type="radio"/> Lambda function
<b>Protocol</b> ⓘ	HTTP ▼
<b>Port</b> ⓘ	80

### Health checks

<b>Protocol</b> ⓘ	HTTP ▼
<b>Path</b> ⓘ	/

#### ▼ Advanced health check settings

<b>Port</b> ⓘ	<input checked="" type="radio"/> traffic port <input type="radio"/> override
<b>Healthy threshold</b> ⓘ	2
<b>Unhealthy threshold</b> ⓘ	2
<b>Timeout</b> ⓘ	5 seconds
<b>Interval</b> ⓘ	10 seconds
<b>Success codes</b> ⓘ	200

- Register Targets
  - Select the instances and **Add to registered**

**Registered targets**  
To deregister instances, select one or more registered instances and then click Remove.

Remove

Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-0ea68c5db78ef0d26	80	running	Security_Group_NV_1	us-east-1a

**Instances**  
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

Add to registered on port 80

Search Instances X

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-0ea68c5db78ef0d26	running	Security_Group_NV_1	us-east-1a	subnet-4750b266	172.31.80.0/20

## Review&Create

**Load Balancer Creation Status**

Successfully created load balancer  
Load balancer [zerbib-loadBalancer](#) was successfully created.  
Note: It might take a few minutes for your load balancer to be fully set up and ready to route traffic, and for the targets to complete the registration process and pass the initial health checks.

**Suggested next steps**

- Discover other services that you can integrate with your load balancer. Visit the [Integrated services](#) tab within [zerbib-loadBalancer](#)
- Consider using AWS Global Accelerator to further improve the availability and performance of your applications. [AWS Global Accelerator console](#)

- In the EC2 console select the newly created load balancer. Write down its **DNS Name** (A Record).

The load balancer's DNS name is : `zerbib-loadBalancer-1551091547.us-east-1.elb.amazonaws.com`

- In the EC2 console select the Target Group. In the lower half of the panel, click on the **Targets** tab. Watch the status of the instance go from **unused** to **initial**.
- Log into the Drupal master instance. Examine the Apache access log `/var/log/apache2/access.log` to see who is connecting to the web server.

## DELIVERABLE 4:

- On your local machine resolve the DNS name of the load balancer into an IP address using the nslookup command (Linux or Windows). Write the DNS name and the resolved IP Address(es) into the report.

```
Server:      192.168.1.1
Address:     192.168.1.1#53

Non-authoritative answer:
Name:   zerbib-loadBalancer-1551091547.us-east-1.elb.amazonaws.com
Address: 54.164.250.69
Name:   zerbib-loadBalancer-1551091547.us-east-1.elb.amazonaws.com
Address: 52.1.94.21

Authoritative answers can be found from:
```

- In the Apache access log identify the health check accesses from the load balancer and copy some samples into the report.

```
172.31.83.54 - - [25/Mar/2020:15:39:11 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:39:21 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:39:25 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:39:31 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
```

[illegible]

```

172.31.83.54 - - [25/Mar/2020:15:42:01 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:42:05 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:42:11 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:42:15 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:42:21 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:42:25 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:42:31 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:42:35 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:42:41 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:42:45 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:42:51 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.23.181 - - [25/Mar/2020:15:42:55 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"
172.31.83.54 - - [25/Mar/2020:15:43:01 +0000] "GET / HTTP/1.1" 200 3440 "-"
"ELB-HealthChecker/2.0"

```

We can see that only two IPs are accessing the Drupal instance, bouncing from one to the other. We can guess that the load balancer is working well and the round-robin algorithm is used.

## TASK 5: LAUNCH A SECOND INSTANCE FROM THE CUSTOM IMAGE

In this task you will launch a second Drupal instance and connect it to the load balancer.

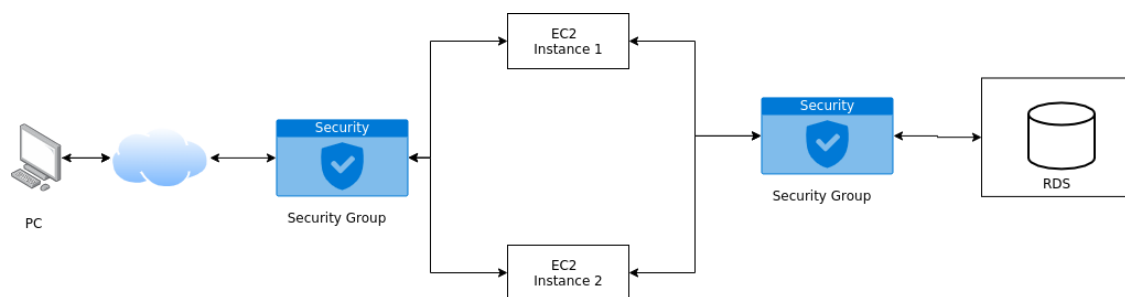
1. Using the custom virtual machine image you created earlier launch a second instance.
2. Make sure that the instance works correctly by navigating with your browser to the Drupal home page of the new instance at `http://hostname/<database-name>/`.
3. Using the AWS console connect the instance to the load balancer. Watch the status of the instance go from **Out of Service** to **In Service**.

Name	Instance ID	Instance Type	Availability Zone	Instance Status	Status Check	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
	i-08756462e69e0...	t2.micro	us-east-1c	running	Initializing	None	ec2-54-85-60-132.compute-1.amazonaws.com	54.85.60.132	-	mint-key-pair...
	i-0ea68c5db78ef...	t2.micro	us-east-1a	running	2/2 check...	None	ec2-3-84-43-137.compute-1.amazonaws.com	3.84.43.137	-	mint-key-pair...

## DELIVERABLE 5:

- Draw a diagram of the setup you have created showing the components (instances, database, load balancer, client) and how they are connected. Include the security groups as well.





- Using the [Simple Monthly Calculator](#) calculate the monthly cost of this setup. You can ignore traffic costs. (Make sure you don't forget to include a component in the calculation. Also don't forget to uncheck the **Free Usage Tier** checkbox at the top.)

Amazon EC2 Service (US East (N. Virginia))	\$ 8.50
Amazon RDS Service (US East (N. Virginia))	\$ 14.75
Amazon Elastic Load Balancing (US East (N. Virginia))	\$ 18.30
AWS Support (Basic)	\$ 0.00
<b>Total Monthly Payment:</b>	<b>\$ 41.55</b>

The price for a month running this setup would be 41.55\$.

## TASK 6: TEST THE DISTRIBUTED APPLICATION

In this task you will test the distributed application with a load generator and use the monitoring tools of the AWS console.

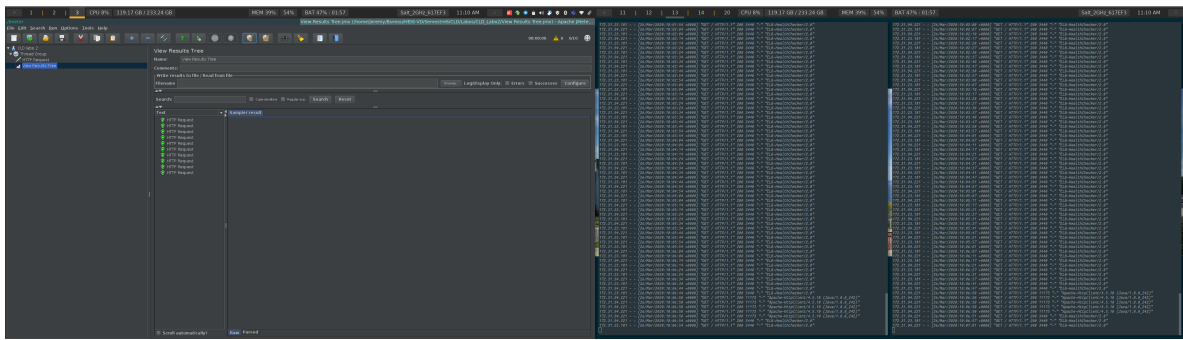
- Download and install on your local machine the JMeter tool from <http://jmeter.apache.org/>.
- Open two terminal windows side-by-side and, using SSH, log into each instance. Bring up a continuous display of the Apache access log by running the command **sudo tail -F /var/log/apache2/access.log**.
- Using the AWS console, enable detailed (1-minute interval) monitoring of the two instances: Select an instance and click on the **Monitoring** tab. Click on **Enable Detailed Monitoring**.
- Follow the instructions on [http://fredpuls.com/site/softwaredevelopment/java/test/test\\_jmeter\\_quick\\_start.htm](http://fredpuls.com/site/softwaredevelopment/java/test/test_jmeter_quick_start.htm) to create a simple test plan. Specify the load balancer as the target for the HTTP requests. Run a test.
- Observe which of the instances gets the load. Increase the load and re-run the test. Observe response times and time-outs. Repeat until you see unacceptable response times and/or time-outs.
- Immediately after having created a high load for the site, re-run the nslookup command to resolve the DNS name of the load balancer into IP addresses to see if there are any changes.

### DELIVERABLE 6:

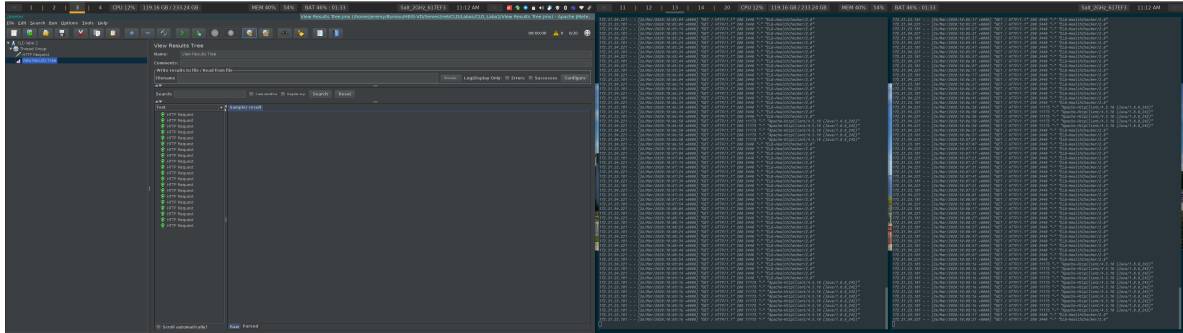
- Document your observations. Include screenshots of JMeter and the AWS console monitoring output.
- When you resolve the DNS name of the load balancer into IP addresses while the load balancer is under high load what do you see? Explain.
- Did this test really test the load balancing mechanism? What are the limitations of this simple test? What would be necessary to do realistic testing?

Note: In this task it is not important that you reproduce exactly an expected behavior of the load balancer. Your load generator (your local machine and your local network) may not behave always the same, the ELB may not always behave the same and you may get results different from your colleagues. **What is important however is that you show that you understand the distributed system that you are testing and that you know how to observe the performance of its components with the monitoring functions provided by AWS. You should show that you are able to correlate the performance with different loads generated by JMeter and draw conclusions.**

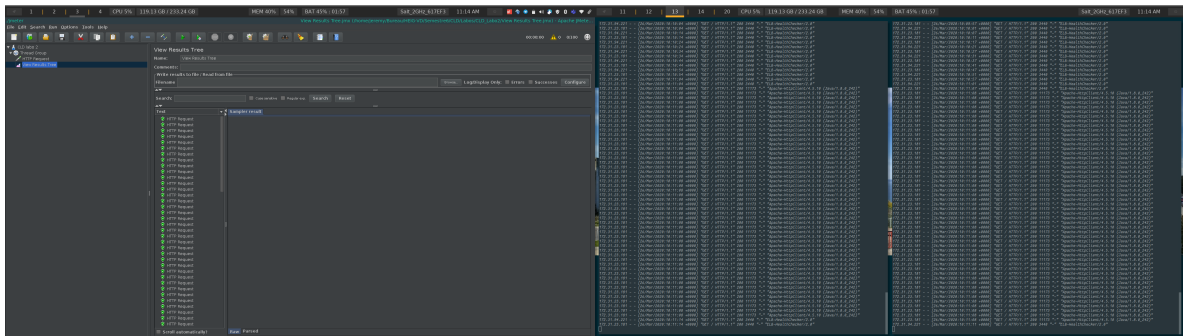
**With 10 threads, 1 loop and a 0 second ramp-up period**



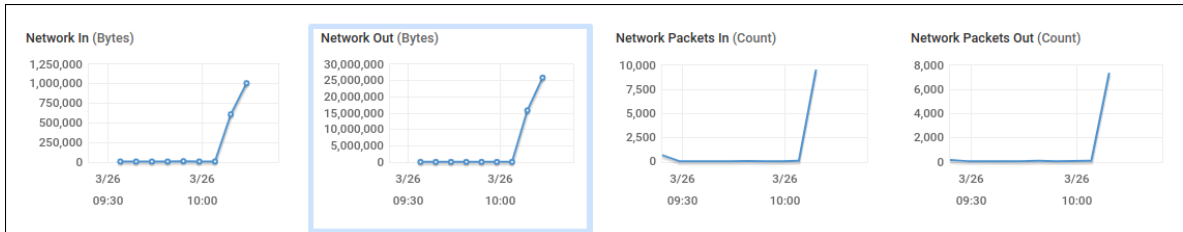
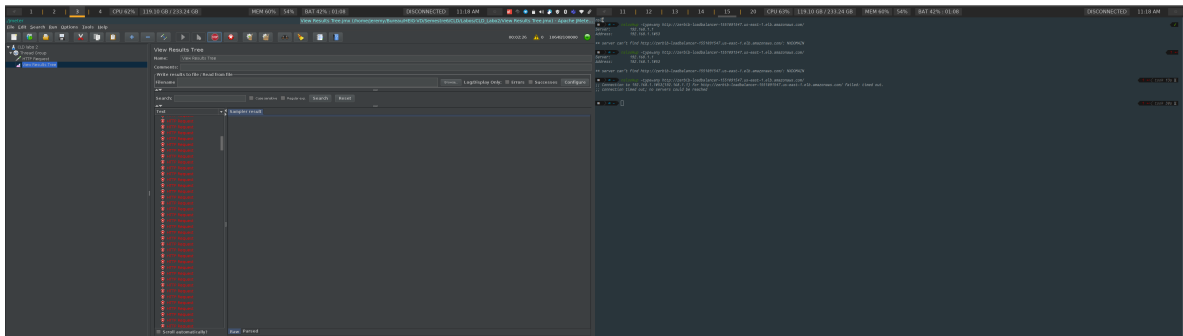
With 20 threads, 1 loop and a 0 second ramp-up period

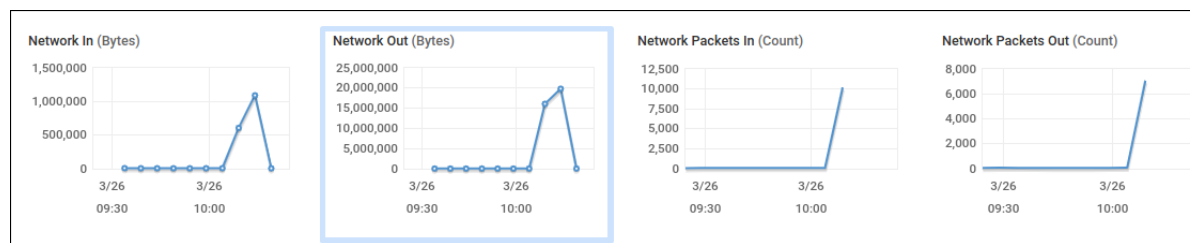


With 100 threads, 1 loop and a 0 second ramp-up period



With 100'000 threads, 1 loop and a 0 second ramp-up period





We can see that this time, the load-balancer is almost drowning in its own load. It looks like the load-balancer is not available and therefore it times out. It is due to the fact that it is overloaded with requests and it cannot accept anymore of them.