(/)

Curriculum

SE Foundations

Average: **121.28%**

0x14. MySQL

DevOps **SysAdmin** **MySQL**

Weight: 1

Project over - took place from Apr 16, 2024 4:00 AM to Apr 17, 2024 4:00 AM

An auto review will be launched at the deadline

In a nutshell...

- **Auto QA review:** 0.0/14 mandatory
- **Altogether: 0.0%**
  - Mandatory: 0.0%
  - Optional: no optional tasks

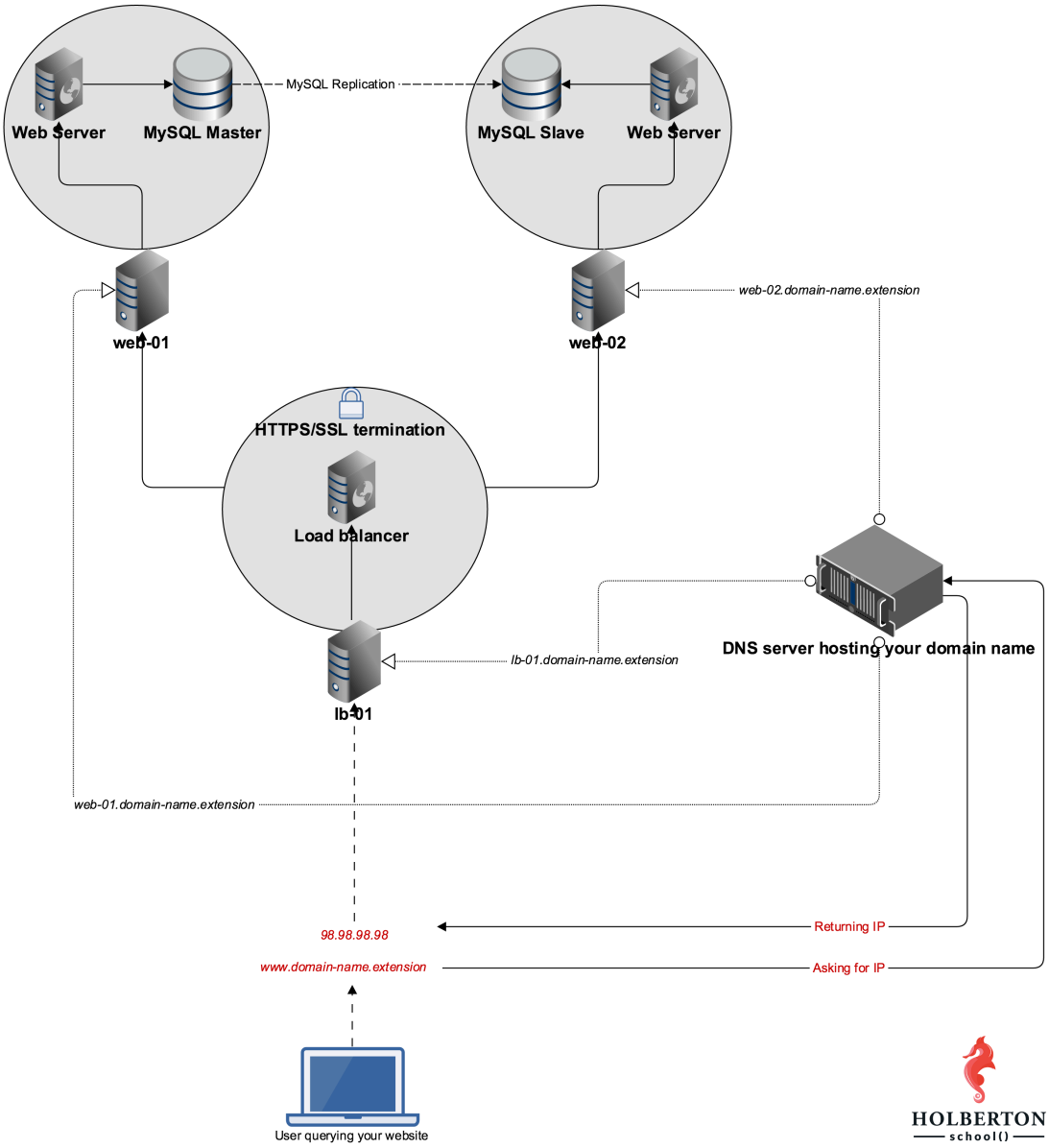
Concepts

For this project, we expect you to look at these concepts:

- [How to :] Fresh Reset and Install mysql 5.7 (/concepts/100002)
- Database administration (/concepts/49)
- Web stack debugging (/concepts/68)

(/)

MySQL



Resources

Read or watch:

- What is a primary-replica cluster (/rltoken/eojqG9FZbA6QVWN5P9cLzA)
- MySQL primary replica setup (/rltoken/z2KVk2UKLMc0RvHMdJmYLg)
- Build a robust database backup strategy (/rltoken/BharnxaLb-BDDYFywzME2Q)

man or help:

- `mysqldump`

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/Lotf0yqq3mNeFHkrW67CZQ), **without the help of Google**:

General

- What is the main role of a database
- What is a database replica
- What is the purpose of a database replica
- Why database backups need to be stored in different physical locations
- What operation should you regularly perform to make sure that your database backup strategy actually works

Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be interpreted on Ubuntu 16.04 LTS
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- All your Bash script files must be executable
- Your Bash script must pass `Shellcheck` (version `0.3.7-5~ubuntu16.04.1` via `apt-get`) without any error
- The first line of all your Bash scripts should be exactly `#!/usr/bin/env bash`
- The second line of all your Bash scripts should be a comment explaining what is the script doing

Your servers

Name	Username	IP	State	
521277-web-01	`ubuntu`	`3.94.181.126`	running	Actions

(/)

Name	Username	IP	State	
521277-web-02	ubuntu	54.158.81.249	running	Actions
521277-lb-01	ubuntu	107.23.116.189	running	Actions

Tasks

0. Install MySQL

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

First things first, let's get MySQL installed on **both** your web-01 and web-02 servers.

- MySQL distribution must be 5.7.x
- Make sure that task #3 (/rltoken/h8QknQcmmLf7oT8esoWgvg) of your SSH project (/rltoken/Wx_BrR5Sk8s3Ywl44-33wg) is completed for `web-01` and `web-02`. The checker will connect to your servers to check MySQL status
- Please make sure you have your `README.md` pushed to GitHub.

Example:

```
ubuntu@229-web-01:~$ mysql --version
mysql  Ver 14.14 Distrib 5.7.25, for Linux (x86_64) using  EditLine wrapper
ubuntu@229-web-01:~$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x14-mysql`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review

1. Let us in!

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

In order for us to verify that your servers are properly configured, we need you to create a user and password for **both** MySQL databases which will allow the checker access to them.

(/)

- Create a MySQL user named `holberton_user` on both `web-01` and `web-02` with the host name set to `localhost` and the password `projectcorrection280hbtn`. This will allow us to access the replication status on both servers.
- Make sure that `holberton_user` has permission to check the primary/replica status of your databases.
- In addition to that, make sure that task #3 (/rltoken/h8QknQcmmLf7oT8esoWgvg) of your SSH project (/rltoken/Wx_BrR5Sk8s3Ywl44-33wg) is completed for `web-01` and `web-02`. **You will likely need to add the public key to web-02 as you only added it to web-01 for this project.** The checker will connect to your servers to check MySQL status

Example:

```
ubuntu@229-web-01:~$ mysql -uholberton_user -p -e "SHOW GRANTS FOR 'holberton_use
r'@'localhost'"
Enter password:
+---------------------------------------------------------------------+
| Grants for holberton_user@localhost                                 |
+---------------------------------------------------------------------+
| GRANT REPLICATION CLIENT ON *.* TO 'holberton_user'@'localhost' |
+---------------------------------------------------------------------+
ubuntu@229-web-01:~$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x14-mysql`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review

2. If only you could see what I've seen with your eyes

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

In order for you to set up replication, you'll need to have a database with at least one table and one row in your primary MySQL server (web-01) to replicate from.

- Create a database named `tyrell_corp`.
- Within the `tyrell_corp` database create a table named `nexus6` and add at least one entry to it.
- Make sure that `holberton_user` has `SELECT` permissions on your table so that we can check that the table exists and is not empty.

(/)

```
ubuntu@229-web-01:~$ mysql -uholberton_user -p -e "use tyrell_corp; select * from nexus6"
Enter password:
+----+-------+
| id | name  |
+----+-------+
|  1 | Leon  |
+----+-------+
ubuntu@229-web-01:~$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x14-mysql`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review

3. Quite an experience to live in fear, isn't it?

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

Before you get started with your primary-replica synchronization, you need one more thing in place. On your **primary** MySQL server (web-01), create a new user for the replica server.

- The name of the new user should be `replica_user`, with the host name set to `%`, and can have whatever password you'd like.
- `replica_user` must have the appropriate permissions to replicate your primary MySQL server.
- `holberton_user` will need SELECT privileges on the `mysql.user` table in order to check that `replica_user` was created with the correct permissions.

```
ubuntu@229-web-01:~$ mysql -uholberton_user -p -e 'SELECT user, Repl_slave_priv FROM mysql.user'
+------------------+-----------------+
| user             | Repl_slave_priv |
+------------------+-----------------+
| root             | Y               |
| mysql.session    | N               |
| mysql.sys        | N               |
| debian-sys-maint | Y               |
| holberton_user   | N               |
| replica_user     | Y               |
+------------------+-----------------+
ubuntu@229-web-01:~$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x14-mysql`

(/)

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review

4. Setup a Primary-Replica infrastructure using MySQL

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

Having a replica member on for your MySQL database has 2 advantages:

- Redundancy: If you lose one of the database servers, you will still have another working one and a copy of your data
- Load distribution: You can split the read operations between the 2 servers, reducing the load on the primary member and improving query response speed

Requirements:

- MySQL primary must be hosted on `web-01` - do not use the `bind-address`, just comment out this parameter
- MySQL replica must be hosted on `web-02`
- Setup replication for the MySQL database named `tyrell_corp`
- Provide your MySQL primary configuration as answer file(`my.cnf` or `mysqld.cnf`) with the name `4-mysql_configuration_primary`
- Provide your MySQL replica configuration as an answer file with the name `4-mysql_configuration_replica`

Tips:

- Once MySQL replication is setup, add a new record in your table via MySQL on `web-01` and check if the record has been replicated in MySQL `web-02`. If you see it, it means your replication is working!

- **Make sure that UFW is allowing connections on port 3306 (default MySQL port) otherwise replication will not work**.

(/)

Example:

web-01

```
ubuntu@web-01:~$ mysql -uholberton_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1467
Server version: 5.5.49-0ubuntu0.14.04.1-log (Ubuntu)

Copyright (c) 2000, 2016, 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> show master status;
+------------------+----------+--------------------+------------------+
| File             | Position | Binlog_Do_DB       | Binlog_Ignore_DB |
+------------------+----------+--------------------+------------------+
| mysql-bin.000009 |      107 | tyrell_corp        |                  |
+------------------+----------+--------------------+------------------+
1 row in set (0.00 sec)

mysql>
```

web-02

```
root@web-02:/home/ubuntu# mysql -uholberton_user -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.49-0ubuntu0.14.04.1-log (Ubuntu)

Copyright (c) 2000, 2016, 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> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 158.69.68.78
                  Master_User: replica_user
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000009
          Read_Master_Log_Pos: 107
               Relay_Log_File: mysql-relay-bin.000022
                Relay_Log_Pos: 253
        Relay_Master_Log_File: mysql-bin.000009
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 107
              Relay_Log_Space: 452
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
```

```
Master_SSL_Verify_Server_Cert: No
(/)             Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
1 row in set (0.00 sec)

mysql>
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x14-mysql`
- File: `4-mysql_configuration_primary, 4-mysql_configuration_replica`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review

5. MySQL backup

mandatory

Score: 0.0% (*Checks completed: 0.0%*)

(https://www.youtube.com/watch?v=ANU-oSE5_hU)

What if the data center where both your primary and replica database servers are hosted are down because of a power outage or even worse: flooding, fire? Then all your data would inaccessible or lost. That's why you want to backup and store them in a different system in another physical location. This can be achieved by dumping your MySQL data, compressing them and storing them in a different data center.

Write a Bash script that generates a MySQL dump and creates a compressed archive out of it.

Requirements:

- The MySQL dump must contain all your MySQL databases
- The MySQL dump must be named `backup.sql`
- The MySQL dump file has to be compressed to a `tar.gz` archive
- This archive must have the following name format: `day-month-year.tar.gz`
- The user to connect to the MySQL database must be `root`
- The Bash script accepts one argument that is the password used to connect to the MySQL database

Example:

(7)

```
ubuntu@03-web-01:~$ ls
5-mysql_backup
ubuntu@03-web-01:~$ ./5-mysql_backup mydummypassword
backup.sql
ubuntu@03-web-01:~$ ls
01-03-2017.tar.gz  5-mysql_backup  backup.sql
ubuntu@03-web-01:~$ more backup.sql
-- MySQL dump 10.13  Distrib 5.7.25, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database:
-- ------------------------------------------------------
-- Server version   5.7.25-0ubuntu0.14.04.1

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Current Database: `tyrell_corp`
--

CREATE DATABASE /*!32312 IF NOT EXISTS*/ `tyrell_corp` /*!40100 DEFAULT CHARACTER SE
T latin1 */;

USE `tyrell_corp`;

--
-- Table structure for table `nexus6`
--

DROP TABLE IF EXISTS `nexus6`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `nexus6` (
  `id` int(6) unsigned NOT NULL AUTO_INCREMENT,
  `firstname` varchar(30) NOT NULL,
  `lastname` varchar(30) NOT NULL,
  `email` varchar(50) DEFAULT NULL,
  `reg_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAM
P,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
/*!40101 SET character_set_client = @saved_cs_client */;
ubuntu@03-web-01:~$
ubuntu@03-web-01:~$ file 01-03-2017.tar.gz
```

```
01-03-2017.tar.gz: gzip compressed data, from Unix, last modified: Wed Mar  1 23:38:
09 2017
ubuntu@03-web-01:~$
```

Repo:

- GitHub repository: `alx-system_engineering-devops`
- Directory: `0x14-mysql`
- File: `5-mysql_backup`

Done? | Check your code | Ask for a new correction | Get a sandbox | QA Review