Internship Experience
BIG DATA ANALYTICS

By
BEEGOO, UMME ZAINAB RUKHSAAR

COMPANY NAME:

OceanDBA Ltd

POSITION:

Junior System and Database Engineer

MENTOR/MANAGER:

Mr Joffrey Michaie

INTERNSHIP PERIOD:

4th November 2019 – 27th March 2020

Agenda

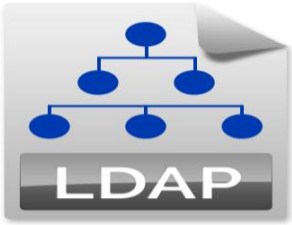
Introduction

Familiarise with Linux

Familiarise with MariaDB

Familiarise with Web Platforms

(R&D) Nginx



(R&D) Nginx + Ldap auth

Skills Acquired

Introduction

- Operating since 2016, OceanDBA has been providing MySQL Consulting, MariaDB Training, RemoteDBA and Support Services to approximately 100 Customers.
- As a Junior System and Database Engineer at OceanDBA Ltd, I have been able to develop my skills in Linux, MySQL, MariaDB and also Nginx mostly.
- Some of the companies trusting OceanDBA are:

Familiarise with Linux

- Linux is a free open source OS based on Unix that can be installed on PCs, laptops, mobiles and more.
- Popular Linux OS distributions include Debian, Ubuntu, Fedora and Red Hat.
- What makes Linux powerful?

-open source

-customization

-security

- System basic commands- Man pages(display the user manual of any command that we can run):

-ls, cd, cp

-pwd: show current position

-less, more, cat: display file contents

-chown: change owner and group of files

-ln -s file link: create symbolic link

-ssh –p port user@host: connect using port p

(ssh root@188.40.116.196 –p)

Familiarise with MariaDB and MySQL

CONSTRAINTS	
NOT NULL	a column cannot have a NULL value
UNIQUE	all values in a column are different
PRIMARY KEY	combination of NOT NULL and UNIQUE
FOREIGN KEY	uniquely identifies a row/record in another table
CHECK	all values in a column satisfies a specific condition

- Installed and Started MariaDB and MySQL

Question Example on MySQL

```
MariaDB [my_questions]> SHOW CREATE TABLE jobs;
+-------+-------------------------------------------------------------
-----------------------------------+
| Table | Create Table
                                    |
+-------+-------------------------------------------------------------
-----------------------------------+
| jobs  | CREATE TABLE `jobs` (
  `job_id` int(11) DEFAULT NULL,
  `job_title` varchar(20) DEFAULT NULL,
  `min_salary` int(11) DEFAULT NULL,
  `max_salary` int(11) DEFAULT NULL CHECK (`max_salary` <= 25000)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 |
+-------+-------------------------------------------------------------
-----------------------------------+
```

Write a SQL statement to create a table named jobs including columns job_id, job_title, min_salary, max_salary and ***check*** whether the max_salary amount exceeding the upper limit 25000.

- **NOTE:**

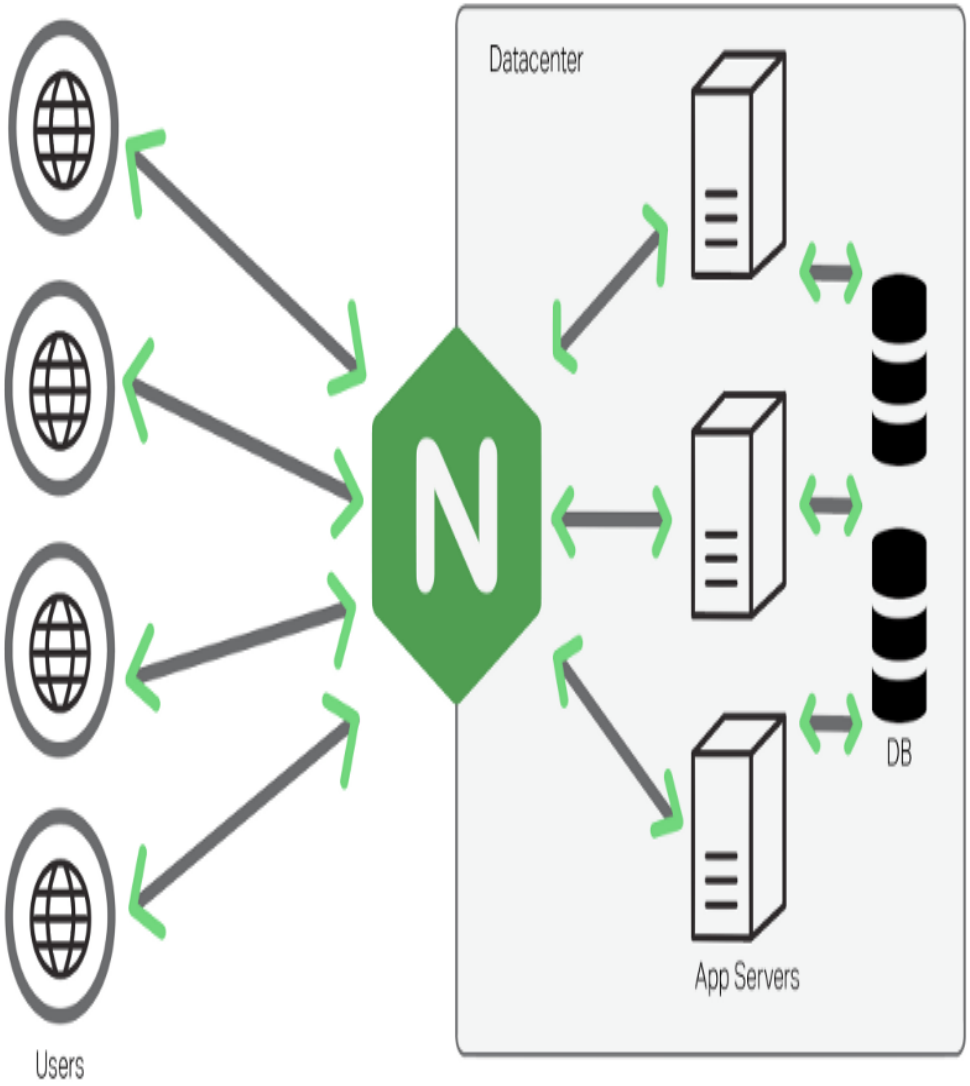
Values(3,“SHOPKEEPER”,8000, 26000) will not be entered into the table as the max salary exceeds the upper limit 25000 and the CHECK constraint will ensure that it satisfies the condition.

```
MariaDB [my_questions]> SELECT * FROM jobs;
+--------+-----------+------------+------------+
| job_id | job_title | min_salary | max_salary |
+--------+-----------+------------+------------+
|      1 | TEACHER   |      12000 |      15000 |
|      2 | MANAGER   |      23000 |      25000 |
+--------+-----------+------------+------------+
2 rows in set (0.034 sec)
```

Familiarise with Web Platforms

- Nginx is a free and open-source web server used to host websites and application of all sizes. It is used as a reverse proxy, HTTP cache and load balancer.
- Installation of Nginx + Php-fpm

1. apt-get install nginx
2. /etc/init.d/nginx start or systemctl start nginx
3. systemctl status nginx (to check if nginx is running)
4. /etc/nginx/nginx.conf (location of main configuration file)



• Created a debian 9 VM from scratch with different IP address to install softwares

• Installation and Configuration of:

- **DoKuWiki** – is an open source wiki program written in PHP that doesn't require a database
- **PhpMyAdmin** – is a free software tool written in PHP, to handle the administration of MySQL over the Web
- **Redmine** – is a free and open source, web-based project management and issue tracking tool. It allows users to manage multiple projects and associated subprojects.

(R&D) Check if Nginx can cache offline pages

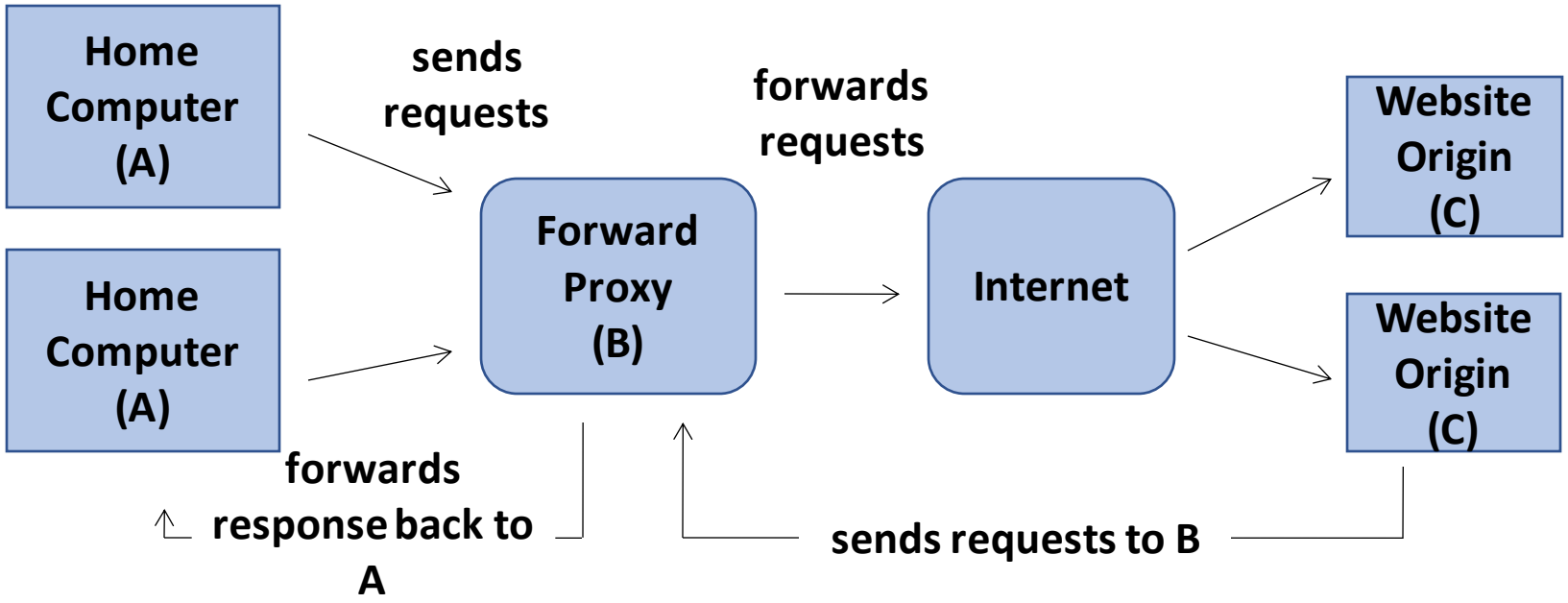
Scenario:

-cache enabled on nginx cache

-backend server nginx works and backend server nginx is stopped

-cache content is still served

-forward and reverse proxy



```
root@debian:/etc/nginx/conf.d# cat re-proxy.conf
#http {
   upstream backend {
      server 10.10.10.10:8091;

   }


   server {

        listen 8090;
        server_name test-back;

        location / {
                    auth_basic "Restricted";
                    auth_basic_user_file /etc/nginx/conf.d/.htpasswd;

                    proxy_pass http://backend;
                    proxy_cache my_cache;

    #ask nginx to ignore the Cache-Control header.
    #cache-control is an http header used to specify browser caching policies in both client requests and server responses.
    proxy_ignore_headers Cache-Control;

    #enforces an expiration for the cached data and is required if ignoring Cache-Control headers.
    #cache response for up to 1 minute.
    #proxy_cache_valid any 1m;

    #cache response for up to 30 minutes.
    proxy_cache_valid any 30m;

    #instructs nginx to deliver stale content when an item is expired or in the process of being updated from the origin server.
    proxy_cache_use_stale error timeout http_500 http_502 http_503 http_504;

                }

        }
```

- This is the reverse proxy server with IP **10.10.10.11** and in this, I have defined which server to include in the load balancing scheme **(10.10.10.10).**

- This server accepts all traffic to port 8090 and passes it to the upstream.
- The **proxy_ignore_headers Cache-Control** will ask nginx to ignore the Cache-Control header.

- The **proxy_cache_valid** sets the caching time for different response codes. Nginx will serve cached data if the gateway times out, thus it will be valid after 30 minutes.

- As for the **proxy_cache_use_stale**, it will allow us to get the expected outcome, that is serving cached content when web server is offline.

• This is the server with IP 10.10.10.10 and I have created a file named php.conf in /etc/nginx/sites-avalaible directory and I have configured the server which is listening on port 8091.

```
root@debian:/etc/nginx/sites-available# cat php.conf
server {

        listen 8091;

        server_name test-back;
        root /usr/share/nginx/html/;
        index index.htm index.php;
   access_log /var/log/nginx/zlab.access.log;
    error_log /var/log/nginx/zlab.error.log;

        client_max_body_size 15M;
        client_body_buffer_size 128K;

        location / {
          try_files $uri /index.php;
          #auth_basic "Restricted";
          #auth_basic_user_file /etc/nginx/.htpasswd;
        }


       location ~ \.php$ {
       try_files $uri =404;

       fastcgi_pass unix:/var/run/php/php7.3-fpm.sock;
       fastcgi_index  index.php;
       include fastcgi_params;
       fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;

       }

}
```

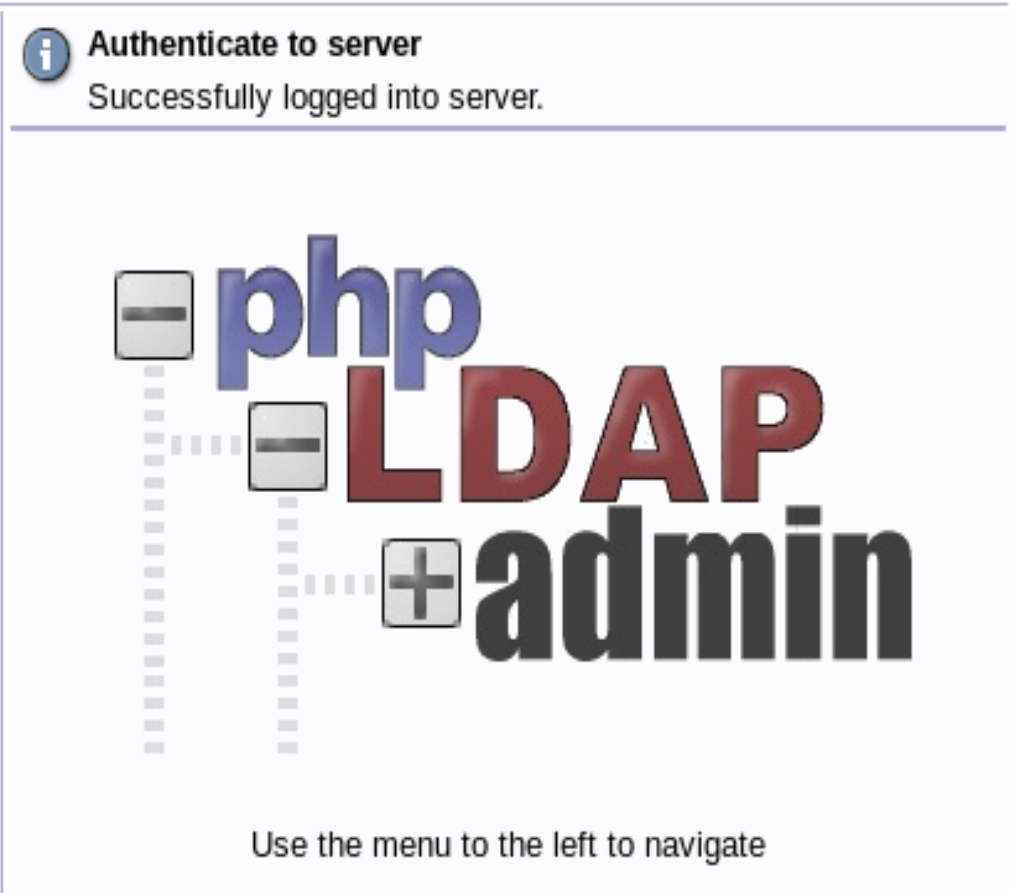
•The proxy_cache_path should also be added in the nginx configuration to have the expected result.

(R&D) Nginx + Ldap auth

Scenario:

-configure nginx for ldap authentication instead of basic auth

-create a htpasswd to restrict access



```
root@debian:/etc/nginx/conf.d# cat reverse-proxy.conf
#http {

   upstream backend {
      server 10.10.10.100:8093;

   }

   server {

        listen 8092;
        server_name test.mu;

        location / {

            proxy_pass http://10.10.10.100:8093;
            #proxy_cache auth_cache;

                  }

    }
```

- The localhost:8093 forwards connections from local port 9999 to localhost:8093 on the server and then we can connect to database.
- I have configured the reverse-proxy with **IP 10.10.10.101** which is listening on **port 8092** and the server name is **test.mu**.
- The **proxy_cache auth_cache** defines a shared memory zone used for caching. The same zone can be used in several places.

```
root@debian:/etc/nginx# cat nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

#the proxy_cache_path directive in the http configuration block allocates 10MB in shared memory for a zone called auth_cache, where metadata is stored.
http {
   proxy_cache_path /var/cache/www levels=1:2 keys_zone=auth_cache:10m max_size=10g inactive=60m use_temp_path=off;
```

- The **proxy_cache_path** directive in the http configuration block, which is in the nginx configuration, creates a local disk directory called **cache**, and allocates 10 MB in shared memory for a zone called **auth_cache**, where metadata is stored.

Skills Acquired

- Being successful in the real world has inspired me to work harder and assume my roles and responsibilities in the particular field.
- Moreover, I have developed my confidence level and working with a team has taught me how to communicate within a group and it was an opportunity for me to apply what I have learned.
- At last, this internship has helped me to gain valuable work experience and real life exposure. Being a part of such group of people has been a unique and promising experience for me and I consider myself lucky to be part of OceanDBA team.


INTERNSHIP
SUCCESS
EXPERIENCE
ADVICE
GOAL
MENTORING
OPPORTUNITY
KNOWLEDGE
PRACTICE
TRAINING
SKILLS

THANK YOU