Clase 21/01/2021

Platform as a Service

Cuando una empresa instala sus sistemas en la nube requieres asi mismo tener acceso a un DBMS
El mas famoso en la actualidad es MySQL.
En un entorno empresarial será necesario llevar a cabo la administración de MySQL, lo cual incluye la isntalación, monitores, respaldo de la BD, recuperación ante desastres, espejo de discos etc.
En este caso la empresa puede contar el personal que se encargue de la administración de la BDMS o contratar el servicio completo en la nube. Este servicio es un ejemplo de plataforma como servicio (PaaS)

Azure Database for MySQL

Permite a la empresa delegar la administración de la base de datos y contar con alta disponibilidad y ecalabilidad dinámica asi como el accesa remoto a MySQL

Creación de un servidor en Azure Database for MySQL

Para crear una base de datos MySQL se deberá ingresar al portal de Azure.
1. Seleccionar la opción "Más servicios"
2. Seleccionar la opción "Todos los servicios"
3. Seleccionar "Servidores de Azure Database for MySQL"
4. Dar click en "Agregar"
5. Seleccionar un grupo de recursos existente o crear uno nuevo.
6. Ingresar el nombre del servidor, por ejemplo: prueba-mysql
7. Seleccionar "Configurar servidor"
8. En la pantalla "Plan de tarifa" se podrá configurar las características del servidor, por ejemplo se podría bajar el número de CPU virtuales de 4 (default) a 2. En la parte derecha de la pantalla se podrá ver el resumen de precios.
9. Una vez configurado el servidor dar click en el botón "Aceptar"
10. Ingresar el nombre del usuario administrador de MysQL, por ejemplo: administrador
11. Ingresar la contraseña del usuario administrador.
12. Dar click en el botón "Revisar y crear"
(Podemos ver el costo estimado al mes 141.85 USD, 2 CPU virtuales, 100 MB de almacenamiento, 7 días de retención de copia de seguridad, redundancia local de copia de seguridad, habilitado el crecimiento automático de almacenamiento)
13. Dar click en el botón "Crear"
14. Dar click en la campana de notificaciones para revisar la implementación en

curso.
15. Cuándo termine la implementación del servidor, dar click en el bóton "Ir al recurso"

Conexión al servidor MySQL

Para conectarnos al servidor de MySQL recién instalado, vamos a ejecutar el monitor de mysql en una computadora:
1. En la parte izquierda de la pantalla seleccionar "Seguridad de la conexión"
2. Dar click en la opción "Agregar IP del cliente"
3. Ingresar en "Nombre de la regla de firewall" el nombre de la regla, por ejemplo: regla1
4. Ingresar en "IP inicial" y en "IP final" la IP de la computadora cliente (la computadora que va a ejecutar el monitor de mysql).
5. Verificar en "Configuración SSL" que SSL esté habilitado.
6. Dar click en el botón "Guardar"
7. En el panel podemos ver el nombre del dominio del servidor, en este caso: prueba-mysql.mysql.database.azure.com
8. Ahora podemos conectarnos al servidor de MySQL ejecutando el monitor de MySQL, en este caso se ejecuta en una computadora con Ubuntu en la cual se ha instalado previamente el monitor de MySQL:

mysql -u administrador@prueba-mysql -p -h prueba-mysql.mysql.database.azure.com --ssl

Como puede verse, la conexión con MySQL se realiza mediante SS

Software as a Service

Como ejemplo se plantea el servicio de envío de correo electrónico que procede la empresa SendGrid a través de Microsoft Azure

SendGrid es un servicio de correo electrónico confiable, escaladle con reportes en tiempo real y fácil de integrar mediante una API

Para enviar un email a través de SendGrid se puede utilizar el protocolo SMTP o bien el Web API v2 de Send Grid

A continuación se muestra un ejemplo de como se envía un email utilizando el Web Api mediante Java:

```
class Correo
{
   static void envia_email(String email, String titulo, String texto) throws Exception
   {
      try
      {
```

```
        HttpsURLConnection httpsURLConnection =
         (HttpsURLConnection)new URL("https://api.sendgrid.com/api/
blocks.get.json").openConnection();
        httpsURLConnection.setRequestMethod("POST");
        httpsURLConnection.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");
        httpsURLConnection.setDoOutput(true);
        String parámetros = "api_user=" + URLEncoder.encode("apikey",
"UTF-8") +
                    "&api_key=" + URLEncoder.encode("API KEY creada por el
usuario", "UTF-8") +
                    "&to=" + URLEncoder.encode(email, "UTF-8") +
                    "&subject=" + URLEncoder.encode(titulo, "UTF-8") +
                    "&html=" + URLEncoder.encode(texto, "UTF-8") +
                    "&from=" + URLEncoder.encode("Username", "UTF-8") +
                    "&fromname=" + URLEncoder.encode("Nombre", "UTF-8");
        DataOutputStream dataOutputStream =
                      new
DataOutputStream(httpsURLConnection.getOutputStream());
        dataOutputStream.writeBytes(parámetros);
        dataOutputStream.close();
        if (httpsURLConnection.getResponseCode() != 200)
        {
          BufferedReader br = new BufferedReader(new
InputStreamReader((httpsURLConnection.getErrorStream())));
          String respuesta;
          while ((respuesta = br.readLine()) != null)
System.out.println(respuesta);
          throw new RuntimeException("Codigo de error HTTP: " +
httpsURLConnection.getResponseCode());
        }
      }
      catch (Exception e)
      {
        throw new Exception("Error al enviar el email: " + e.getMessage());
      }
    }
}
```