PROFESSIONAL DATA ENGINEER - *Google Cloud Platform*

TITLE: "Introduction to Google Cloud - Cloud SQL Allow whitelisting"

Author:

- *Name: "Vignesh Sekar"*
- *Designation: "Multi Cloud Architect"*
- *Tags: [Google Cloud, DataEngineer, Python, PySpark, SQL, BigData]*

There are different ways you can connect to a Cloud SQL instance:

1. The Cloud Proxy
2. */Public IP with allowlisted IPs of consumers/*
3. Private IP from VPC connected applications
4. Language-specific connectors (e.g. JDBC socket factory for Java)
5. Writing your own connector using API calls for ephemeral SSL creation/connection
6. Setting up SSL encryption with permanent key pairs

Public IP with allowlisted IPs of consumers

- This lab is going to walk you through, step-by-step instructions on how to connect to your Cloud SQL instance using the public IP option, and allowing only specific IP addresses to connect.

- This is arguably the least secure way to create and use a Cloud SQL instance.

- If you're SSH'd in, using DNS lookup with something like dig is also a good way

  - dig @resolver1.opendns.com ANY myip.opendns.com +short -4

- Once you have the IP address, put it into the Network box to whitelist that specific connection.

- To allow only one IP address, just put the address, or use this format: 117.248.13.150/32 * E.g. 104.132.11.92 or 104.132.11.92/32 are both equivalent and only allow that specific IP address

- To allow a range of IP addresses: * 104.132.11.0/24 means it will allow 104.132.11.0 to 104.132.11.255

- The number after the slash refers to how much of the IP address to use as a filter.

- Click Done once you've added any IP addresses you want to allow

- Click the Create button and you'll be returned to the instance list and should see your instance

- Easiest way to verify connectivity is by using something like psql: * psql "host= port=5432 sslmode=disable user=postgres" and then enter the password you specified at database creation * **NOTE** that even though setting the sslmode=disable, the Google SQL proxy provides the encrypted connection

```
 * psql -h 34.150.173.226 -U postgres

 * psql host=34.150.173.226 port=5432 sslmode=disable user=postgres

 * gcloud sql instances patch mypostgresinstance --authorized-networks=34.124.130.8
```