

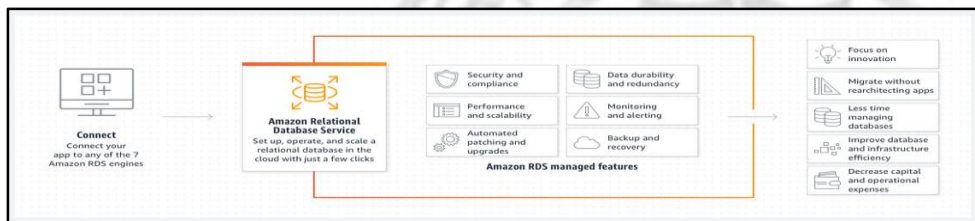
Experiment-7

Aim: To know the concept of Database as a service running on cloud and to demonstrate the CRUD operations on different SQL and NOSQL databases running on cloud like AWS RDS, Azure SQL/Mongo Lab/Firebase.

Prerequisites: Firebase registration

Theory :- Amazon RDS

Amazon Relational Database Service (RDS) is a collection of managed services that makes it simple to set up, operate, and scale databases in the cloud. It supports - Amazon Aurora with MySQL compatibility, Amazon Aurora with Postgre SQL compatibility, MySQL, Maria DB, Postgre SQL, Oracle, and SQL Server — and deploy on-premises with Amazon RDS on AWS outposts.



Amazon RDS makes it easy to use replication to enhance availability and reliability for production workloads. Using the Multi-AZ deployment option, you can run mission-critical workloads with high availability and built-in automated fail-over from your primary database to a synchronously replicated secondary database. Using Read Replicas, you can scale out beyond the capacity of a single database deployment for read-heavy database workloads.

Benefits

Lower administrative burden

You can use the AWS Management Console, the Amazon RDS Command Line Interface, or simple API calls to access the capabilities of a production-ready relational database in minutes.

Amazon RDS database instances are pre-configured with parameters and settings appropriate for the engine and class we have selected. You can launch a database instance and connect your application within minutes. DB Parameter Groups provide granular control and fine-tuning of databases.

Performance

Amazon RDS General Purpose Storage is an SSD-backed storage option delivers a consistent baseline of 3 IOPS per provisioned GB and provides the ability to burst up to 3,000 IOPS above the baseline. This storage type is suitable for a broad range of database workloads.

Scalability

As your storage requirements grow, you can also provision additional storage. The Amazon Aurora engine will automatically grow the size of your database volume as your database storage needs grow, up to a maximum of 64

TB or a maximum you define. The MySQL, MariaDB, Oracle, and PostgreSQL engines allow you to scale up to 64 TB of storage and SQL Server supports up to 16 TB. Storage scaling is on-the-fly with zero downtime.

Availability and durability

Database snapshots are user-initiated backups of your instance stored in Amazon S3 that are kept until you explicitly delete them. You can create a new instance from database snapshots whenever you desire. Although database snapshots serve operationally as full backups, you are billed only for incremental storage use.

Security

Amazon RDS allows you to encrypt your databases using keys you manage through AWS Key Management Service (KMS). On a database instance running with Amazon RDS encryption, data stored at rest in the underlying storage is encrypted, as are its automated backups, read replicas, and snapshots.

Cost-effectiveness

There is no up-front commitment with Amazon RDS; you simply pay a monthly charge for each database instance that you launch. And, when you're finished with a database instance, you can easily delete it. To see more details, visit the Amazon RDS Instance Types page and the Amazon RDS Pricing page.

Similar to AWS RDS, Firebase by Google also offers practical and efficient database solution.

Firebase

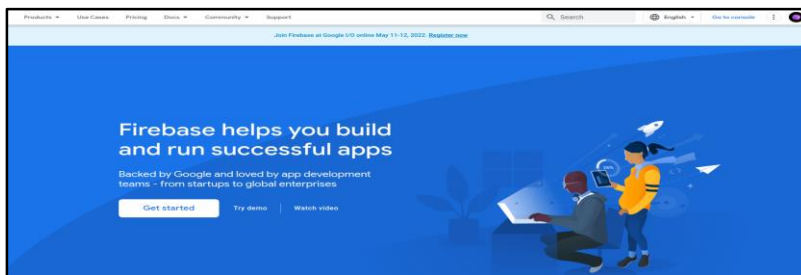


Firebase is a mobile platform meant for app development. Firebase offers multiple services such as Realtime database, Firestore, Authentication and Cloud messaging to simplify and automate backend tasks.

Cloud Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, you store data in *documents*, which are organized into *collections*. Each *document* contains a set of key-value pairs. Cloud Firestore is optimized for storing large collections of small documents. All documents must be stored in collections. Documents can contain *subcollections* and nested objects, both of which can include primitive fields like strings or complex objects like lists. Collections and documents are created implicitly in Cloud Firestore. Simply assign data to a document within a collection. If either the collection or document does not exist, Cloud Firestore creates it.

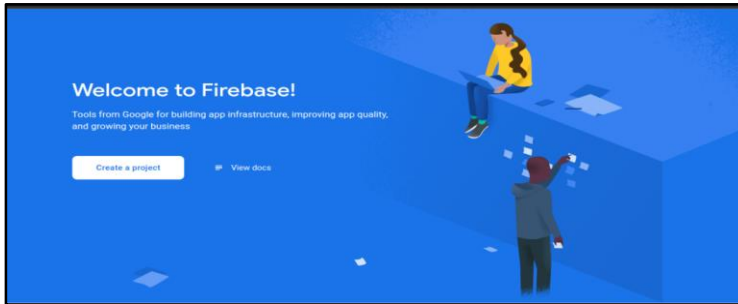
Registering to Firebase

Users register to Firebase using a standard Google account. Once registered, users directly land on the Firebase dashboard where they may create projects.

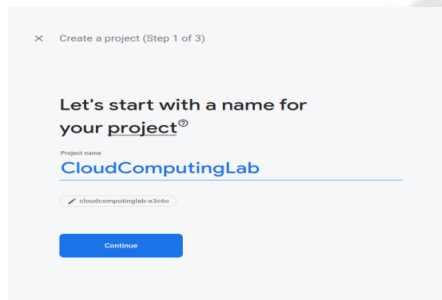


Creating a project

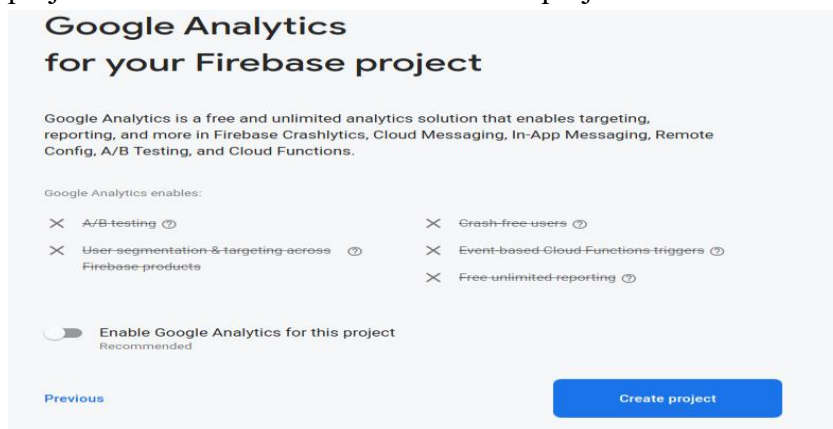
1. To create a project in Firebase, go to the Firebase Console.



2. Give your project a name.



3. For the purposes of this practical, Firebase analytics may be disabled. Disable it and click on Create a project. Wait for Firebase to initialize the project.



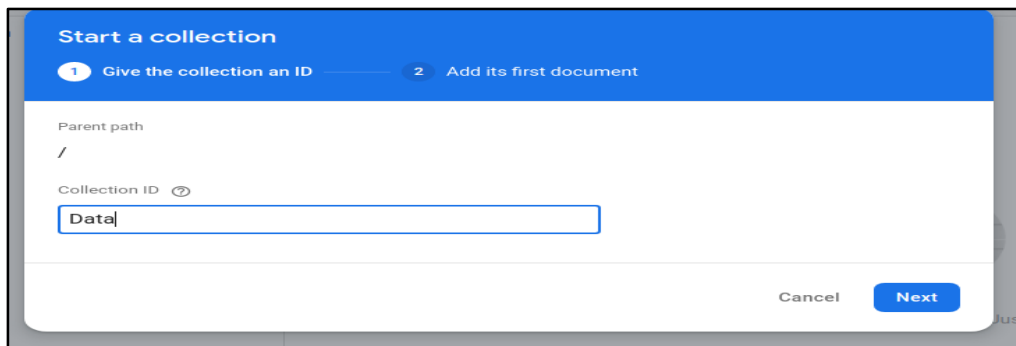
4. Once the project is initialized, the standard Firebase project dashboard can be seen. Head over to the Firebase Firestore, under the Build section in the left panel.
5. Create a Firestore database. Start the database in **test mode** and keep the default location.



CRUD operations on the database

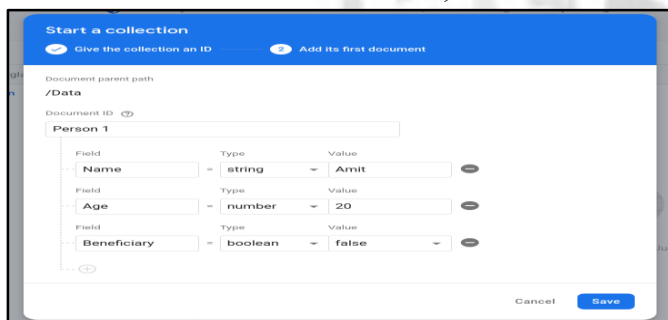
1. Create operation

Click on *Start a collection*. Assign a Collection ID of your choice. This will act as a container for all of the documents created within.



Once a Collection is created, Firebase will give a prompt for creating the first document of the Collection. We can create a document ID with an arbitrary name or auto generate the ID. The fields of a document require a name, type and a default value. Fields are created according to the project's requirements.

Once the first document is created, more documents can be created with *Add Document*.



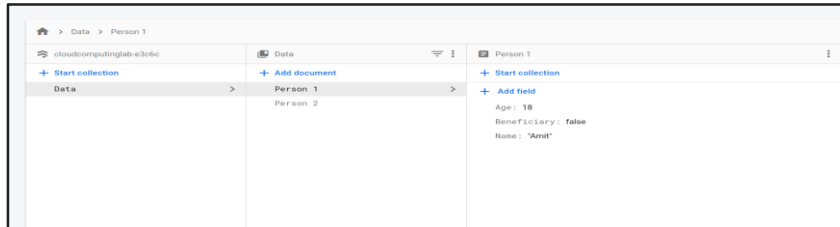
2. Update operation

If any of the data needs to be updated, it can be done manually by clicking on the pencil icon (Edit Field) to the right of each field.



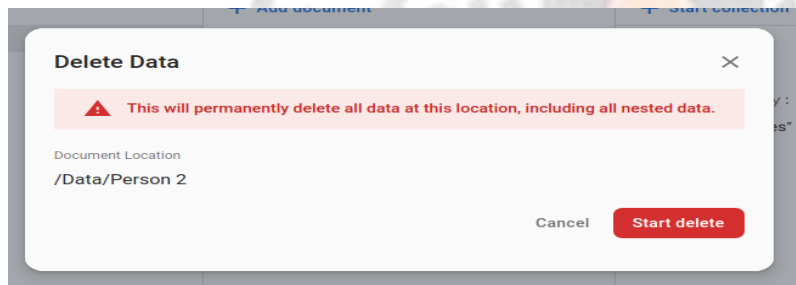
3. Read operation

Once the database is integrated into an application, the application can use Firebase SDK's library methods to fetch data from a collection or a document. The application can filter a specific document or a specific field, as needed. If the database needs to be viewed manually, it can be done by going to the Firestore, where we can get an overview of the data.



4. Delete operation

In Firebase, entire collections, documents or fields can be erased. To delete a document or a collection, click on the kebab menu (3 vertical dots) to the right of the document/collection. Click on "Delete document" (or collection). A prompt will be displayed which asks us to confirm the deletion by typing the entity's name. This is not needed for a document



Once deleted, the document/collection is found to be erased from the database.

Conclusion :-

Sign and Remark:

R1 (3 Marks)	R2 (2 Marks)	R3 (5 Marks)	R4 (5 Mark)	Total (15 Marks)	Signature