ature is in a pre-release state and might change or have limited support. For more information, see the product launc
(/products/#product-launch-stages).

With Cloud Functions, you can handle events in the Firebase Realtime Database with no need to update client code. Cloud Functions lets you run database operations with full administrative privileges, and ensures that each change to the database is processed individually. You can make Firebase Realtime Database changes via the Firebase Admin SDK (https://firebase.google.com/docs/database/admin/start).

f you are only using Firebase products in your application, you may want to use the Cloud Functions for Firebase SD
://firebase.google.com/docs/functions/) as it performs some convenience operations that simplify interacting with
nt Firebase products.

In a typical lifecycle, a Firebase Realtime Database function does the following:

1. Waits for changes to a particular database location.
2. Triggers when an event occurs and performs its tasks.
3. Receives a data object that contains a snapshot of the data stored in the specified document.

Functions lets you handle database events at two levels of specificity; you can listen specifically for only creation, update, or deletion events, or you can listen for any change of any kind to a path. Cloud Functions supports the following event types for the Realtime Database:

Event Type	Trigger
`providers/google.firebase.database/eventTypes/ref.write`	Triggered on any mutation event: when data is created, updated, or deleted in the Realtime Database.
`providers/google.firebase.database/eventTypes/ref.create`	Triggered when new data is created in the Realtime Database.

`providers/google.firebase.database/eventTypes/ref.update`	Triggered when data is updated in the Realtime Database.
`providers/google.firebase.database/eventTypes/ref.delete`	Triggered when data is deleted from the Realtime Database.

To control when and where your function should trigger, you need to specify a path, and optionally specify a database instance.

Path specifications match *all* writes that touch a path, including writes that happen anywhere below it. If you set the path for your function as `/foo/bar`, it matches events at both of these locations:

In either case, Firebase interprets that the event occurs at `/foo/bar`, and the event data includes the old and new data at `/foo/bar`. If the event data might be large, consider using multiple functions at deeper paths instead of a single function near the root of your database. For the best performance, only request data at the deepest level possible.

You can specify a path component as a wildcard by surrounding it with curly braces; `foo/{bar}` matches any child of `/foo`. The values of these wildcard path components are available within the `event.params` object of your function. In this example, the value is available as `event.params.bar`.

Paths with wildcards can match multiple events from a single write. An insert of:

matches the path `/foo/{bar}` twice: once with `"hello": "world"` and again with `"firebase": "functions"`.

When using the Cloud Console, if you do not specify an instance, the function deploys to the default database instance for your project.

When using the `gcloud` command-line tool, the instance must be specified as part of the `--trigger-resource` string. Usually, the default instance name is the same as your project ID.

This rule may not apply to projects that were created in the legacy Firebase console and later migrated to GCP.

For example if you are deploying a function in a project with the ID `my-project-id`, you would use the following in your `--trigger-resource` string:

When handling a Realtime Database event, the `data` object contains two properties that are provided in JSON object format:

- `data`: a snapshot of the data taken prior to the event that triggered the function.
- `delta`: a snapshot of the data taken after the event that triggered the function.

the precise structure of these objects will depend on the structure of your database itself.

The following gcloud command deploys a function that will be triggered by `create` events on the path `/messages/{pushId}/original`:

Argument	Description
`--trigger-event` *NAME*	The name of the event type that the function wishes to receive. In this case, it will be one of the following: write, create, update or delete.
`--trigger-resource` *NAME*	The fully qualified database path to which the function will listen. This should conform to the following format: `projects/_/instances/`*DATABASE_INSTANCE*`/refs/`*PATH*.
`--runtime` *RUNTIME*	The name of the runtime you are using. For a complete list, see the **`gcloud`** reference (/sdk/gcloud/reference/functions/deploy#--runtime).