GOOGLE CLOUD PLATFORM

CLOUD STORAGE

OBJECT LIFECYCLE MANAGEMENT

- Automate common tasks with a Lifecycle Management Configuration
- Configuration contains a set of rules which apply to current and future objects in the bucket.
- Each rule should contain only one action; you can specify a set of conditions for any action
- If you specify multiple conditions in a rule, an object has to match *all* of the conditions

- Actions are not necessarily performed right away by Cloud Storage
- You can track Lifecycle actions with logs and by enabling Cloud Pub/Sub Notifications for Cloud Storage for your bucket

- **Lifecycle actions -** Delete, SetStorageClass
- **Lifecycle conditions -** Age, CreatedBefore, IsLive, MatchesStorageClass, NumberOfNewerVersions

DEMO: OBJECT LIFECYCLE

To enable lifecycle management for a bucket:

1. Open the Cloud Storage browser in the Google Cloud Platform Console
2. In the bucket list, find the bucket you want to enable, and click None in the Lifecycle column
3. Click Add rule in the lifecycle rules page
4. Select the condition(s) under which an action is taken
5. Select the action to take when an object meets the condition(s)
6. Click Save

To check the lifecycle configuration set on a bucket:

```
$ gsutil lifecycle get gs://[BUCKET_NAME]
```

OBJECT VERSIONING

- Cloud Storage offers the Object Versioning feature to support the retrieval of objects that are deleted or overwritten
- When Object Versioning is enabled, you can list archived versions of an object, restore the live version of an object to an older state, or permanently delete an archived version
- Versioning can add costs but can be managed with Lifecycle control
- Cloud Storage uses two properties that together identify the version of an object:
  - The version of the object's data
  - The version of the object's metadata

RETENTION POLICY & BUCKET LOCKS

- You can add a retention policy to a bucket to specify a retention period
  - If a bucket has a retention policy, objects in the bucket can only be deleted or overwritten once their age is greater than the retention period
  - A retention policy retroactively applies to existing objects in the bucket as well as new objects added to the bucket

- You can lock a retention policy to permanently set it on the bucket
  - Once you lock a retention policy, you cannot remove it or reduce the retention period
  - You cannot delete a bucket with a locked retention policy unless every object in the bucket has met the retention period
  - You can increase the retention period of a locked retention policy
  - Locking a retention policy can help your data comply with record retention regulations
- You can place *holds* on individual objects which prevent them from being deleted or overwritten

STATIC WEBSITE HOSTING

- We can use buckets to host static websites
- The name of the bucket needs to be the same as the cname record
- The objects inside the bucket are the webpages for your site

  Example: http://[BUCKET_NAME]/[OBJECT_NAME]

  http://www.example.com/index.html

- You can either make all files in your bucket publicly accessible, or you can set individual objects to be accessible
- You can set properties to the bucket to assign a page as the default and for errors

  Example – use MainPageSuffix to set index.html & NotFoundPage to set 404.html

- When hosting static assets for a dynamic website, you do not need to create a CNAME record and point to a bucket with the same name as you do for a static website.

DEMO: STATIC WEBSITE HOSTING

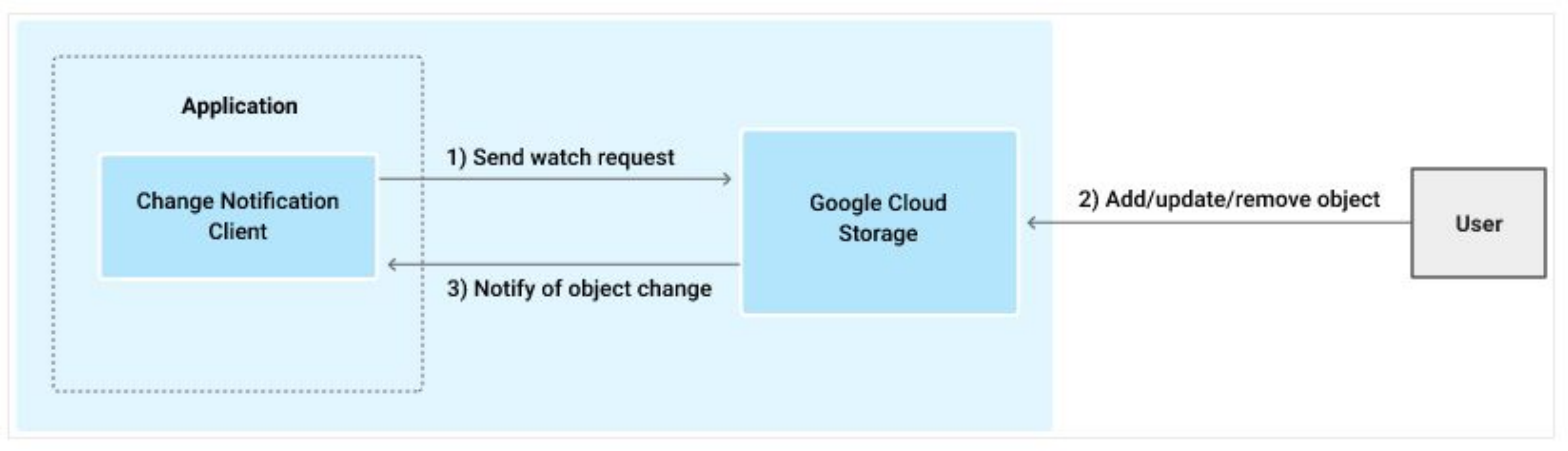
1. Open the Cloud Storage browser in the Google Cloud Platform Console.
2. Create a bucket with the name of the url
3. Select the bucket and upload files from the objects tab
4. Check permissions
5. Assign specialty pages
6. Test your site
7. If you have a domain, you can point it to: c.storage.googleapis.com.

NOTIFICATIONS

- Object Change Notification can be used to notify an application when an object is added/updated/deleted in a bucket.

  E.g. Add a new picture to a bucket, an application could be notified to create a thumbnail.
- To start watching a bucket for change notification events, you can use this command:

  $ gsutil notification watchbucket [-i *ChannelId*] [-t *ClientToken*] *ApplicationUrl* gs://*BucketName*
- This will create a notification channel that sends notification events to the given application URL for the given bucket.


Application
Change Notification Client
1) Send watch request
3) Notify of object change
Google Cloud Storage
2) Add/update/remove object
User

NOTIFICATIONS

- Cloud Pub/Sub Notifications sends information about changes to objects in your buckets to Cloud Pub/Sub
- The information is added to a Cloud Pub/Sub topic of your choice in the form of messages
- Implemented by adding a *notification configuration* rule to a bucket that specifies the topic, trigger events and notification details
- Takes up to 30 seconds to begin sending notifications, guarantees at-least-once delivery to Cloud Pub/Sub

- Cloud Functions: If you only want to trigger a lightweight, stand-alone function in response to events
- Allow you to execute JavaScript functions when an object in your bucket changes
- Your buckets must reside in the same project as Cloud Functions

DEMO: OBJECT VERSIONING

1. Upload file to a Cloud Storage Bucket
2. Take note of the generation number and the metageneration number
3. Enable versioning: $ gsutil versioning set on gs://[BUCKET_NAME]
4. Change metadata and take note of the generation/metageneration number
5. Upload new version and take note of the generation/metageneration number
6. Delete live version
7. Restore an archived version
8. Disable versioning