**Creating an RDR Environment**

As of now, raw-data-repository (RDR) is a Google App Engine application that must be run in App Engine Standard Python 3.7.

These are the steps to stand up a new RDR environment:

1. Create a Google Cloud Project (PROJECT) for the environment
2. Enable Cloud SQL Admin API
3. Create required service accounts
   * + See prod\_service\_accounts.txt
4. Create dedicated cloud storage buckets for the environment (as needed)
   * See prod\_buckets\_list.txt
5. Create BigQuery rdr\_ops\_data\_view dataset
   * either via the web interface or with bq mk --dataset rdr\_ops\_data\_view
6. Update config and tooling files
   * Create config: “current\_config.<isodate>.json” file at gs:// pmi-drc-api-test/app\_engine\_configs/all-of-us-rdr-<PROJECT>
   * Update rdr\_service/tools/tool\_libs/config\_editor.py to handle new project.
   * Create rdr\_service/cron\_<PROJECT>.yaml to override any settings from cron\_default.yaml (if needed)
   * Update rdr\_service/tools/auth\_setup.sh to handle the new project.
   * Update rdr\_service/tools/tool\_libs/app\_engine\_manager.py to handle the new project.
   * Update rdr\_service/services/gcp\_config.py with new project
   * Update rdr\_service/services/gcp\_db\_daemon.py with new project
7. commit changes and create local tag (TAG) <PROJECT>-initial. *Do****not****push this tag to github.*
8. From Google Cloud Console, Create new App Engine application (just enable it for the project)
9. run rdr\_service/tools/tool\_libs/app\_engine\_manager.py to deploy application to the new project.
10. Create Cloud Task Queues:

Graphical user interface, table

Description automatically generated with medium confidence

1. Create PubSub Topics:

Graphical user interface, text, email

Description automatically generated

1. Create PubSub notifications
   * See prod\_pub\_sub\_notifications.txt
2. Deploy Cloud Functions:

Graphical user interface, application, table

Description automatically generated