**Created 7/17/2020**

**Test Prep tips:**

* Set your test date up **THEN** study. Mallika and I suggest a date 1-3 months out from now to take your test. For me I was like a rocket the first month. Got a little bored in month 2, finally got back on track in month 3 after I scheduled the test and I had a "due date".
* Register for the exam here: <https://www.webassessor.com/home.do?page=CANDIDATE&tabs=1&branding=GOOGLECLOUD>
* Google Courses is worth taking for the Qwiklabs alone. I don’t know if I’d recommend this in addition to Linux Academy however. If you have access to both I suggest focusing on Linux Academy but completing the Qwiklabs as part of your preparation.
* DO sign up for Linux Academy. Explanations are great (far better than coursera/Google Courses). Seems to balance "learning the test" with "learning the concepts" pretty well.
* Notes from Linux Academy classes. Even if you don't pay for the subscription ($50/mo) these notes are freely available and worth the read. Lots of exam tips in here as well. <https://interactive.linuxacademy.com/diagrams/MasterBuildersGuide.html>
* You should know the case studies in the link above. I suggest memorizing the diagrams for MountKirk and Terrafirm.
* Learn the keywords (see below). You'll see the benefit in the test taking section.
* Take the official practice exam at least twice: <https://cloud.google.com/certification/practice-exam/cloud-architect>
* Take the Linux Academy practice test at least a handful of times. The first time around I was at 60%. With 3 focused days on study and a couple more evenings I was in the 90+% category.
* If you learn by flow chart here's a great place to start: <https://grumpygrace.dev/posts/gcp-flowcharts/#storage-and-data> It’s the best way to learn the material. However, I ultimately didn't use it. I instead used the keywords strategy below.
* My best guess is I got 85% +/- 5% of the questions right. I have no idea what was required to pass the test. I suspect I over studied/stressed about getting as close to 100% as possible. It's hard when you have ZERO idea what is required to pass it.
* People ask me how hard the test was. It was VERY hard but I was doing everything I could to get 100% not knowing passing grade. I'd say questions break down as follows:
  + 20% - Came right out of Linux Academy and GCP practice exams. Make sure you take those tests enough you've seen every question in the test bank! (Amy says closer to 40% of hers)
  + 30% - Were straight forward material I learned or the elimination method made it clear what the right answer was.
  + 25% - Were moderately hard but the elimination method allowed me to make the most likely guess.
  + 25% - Were VERY hard. 3 I had literally no idea despite the good amount of study I did. Another handful were so hard even after elimination I was mostly guessing.

**Keywords:** When trying to understand scenario requirements it helps to break the question down to keyword that define the solution.

NOTE: Keep in mind these keywords represent the most likely correct answer accounting for cost, complexity, training, best of breed, etc. You will get a few questions that don’t include the best option. You should know the 2nd best option even if it’s not presented below.

* **NoSQL** - Doesn't mean it doesn't use SQL query language but instead refers to being NON-RELATIONAL. Generally speaking, NoSQL doesn’t use SQL language either.
* **Transactional NoSQL database** = Cloud Datastore. BigTable can be an option also but generally wrong because it's much more expensive.
* **Timeseries database** - Means either Cloud BigTable or Cloud BigQuery.
  + BigQuery for more efficient storage
  + BigTable for lower response time.
  + BigQuery is generally seen as an analytics tool whereas BigTable is not. However, you can query BigTable data vial BigQuery too. Generally, won't be the case though.
* **Query data > 10 TB** - Can't be CloudSQL . . . BigQuery instead.
* **Query data < 10 TB and single region** - CloudSQL is an option.
* **Streaming/arrives late/Buffers data** - Pub/Sub
* **Hardened Linux** - Always associated managed instance group.
* **Dynamically scale** - That’s generally everything. Ignore this requirement.
* **Analyze and optimize for performance** - Stackdriver.
* **Automation framework** - Deployment manager (or Terraform)
* **Hadoop/Spark** - Dataproc
* **Convert onprem HDFS** - Cloud Storage first HDFS second
* **Block Storage** - Persistent Disk (can't use Cloud storage it doesn't support block level)
* **Containers** - Kubernetes
* **Network protocols other than HTTPS** - Kubernetes or Compute Engine
* **Websites** - App Engine. . . Web facing code such as websites and mobile apps. Only supports certain languages.
* **Event Driven** - Cloud functions
* **Relational Data** - Cloud SQL or Spanner
* **Analytics Database** - BigQuery
* **In memory** - Cloud Memorystore (Redis)
* **No-Ops/managed servic**e - No infrastructure to manage such as GKE and App Engine.
* **Batch** - Think Cloud Storage
* **Data Lake/HDFS** - Cloud Storage
* **User Profile and Game State** = Cloud Datastore. BigTable (the other NoSQL DB) possible if Cloud Datastore doesn't exist.
* **BigQuery Federated Datasources** = BQ External Datasources.
* **Load testing** - "dd" (linux OS tool), ap (or ab (is apache tool) VERIFY THIS

**Other quick hit explanations:**

* **Dataproc vs. Dataflow for ETL** - Dataproc when already using hadoop or a DevOps approach. Dataflow for hands off server-less approach.
* **Dataflow** - Beam. Build data transformation pipelines.
* **Dataprep** - built on dataflow by providing easy UI. Ex. Consume data see suggestions on how to transform, enhance, etc. then send to Dataflow for ultimate transformation. It helps you build Dataflow pipelines essentially.
* **Data Lab vs. Data Studio** - Data Lab = Jupyter notebooks for analyzing data (changed to AI Platform Notebook in Big Data Engineer exam). Data Studio = tooling around generating dashboards including other managed services to get it to that state.
* **Instance template vs. Compute Engine Template** - Use instance template if few changes from base. Use CET if there are a lot of changes and/or need quick spin up required. It takes longer to get a CET setup.
* **Storage IAM roles** - Cloud Storage Standard role (scope = bucket and project) vs. Cloud Storage Legacy Roles (scope = bucket only) vs. ACLs Object level (scope = bucket or object but inheritable)
* Deleted object in Cloud Storage is permanent unless object level versioning is enabled.
* **IAM roles** - Primitive roles (bad) vs. predefined roles (good).
* **App Engine Standard vs. Flexible** - Standard. . . Scale up or down infinitely, scale up/down (seconds), limited supported applications. Flexible. . . .more languages & RPMs, scale up/down (minutes), remote VPC/VPN access. Both are often used with Datastore.
* **Cloud Firestore** - Know that its replacing Cloud Datastore.
* **Images** - CE Images are required for making instance templates and instance groups. Snapshots are not really an option and they will try to trick you on this during the test.
* Remember you must stop the VM before creating the Image
* **Images vs. snapshots** - Images are for instance templates, etc. Snapshots are for backups. Snapshots aren't accessible outside the zone. Images are.
* **Resizing Kubernetes** - Can increase/decrease the number of nodes (workers) but NOT re-size the nodes. VERIFY THIS

**App Engine Comparison:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Standard** | **Vs** | **Flexible** |
| **Constraints** | More constraints |  | Less constraints, more customization, but more management |
| **Languages** | Python, Java, PHP, Go, NodeJS |  | Python, Java, PHP, Go, NodeJS, Ruby or .NET or any other in own docker container. |
| **Scaling speed** | Faster scale up (seconds) |  | Slower scale up (minutes), consistent traffic, gradual scale up/down |
| **Cost** | Pay for what you use (as little as zero) |  | Always costs something |
| **File system access** | NONE (Sandbox) |  | Available |
| **OS Access** | NONE (Sandbox) |  | SSH |
| **3rd Party Binaries** | N/A |  | Yes |
| **External access** | N/A |  | VPC/VPN connections (ex. Connect from App Engine to on-prem DB via VPN) |
| **Virtualization Model** | ??? |  | Docker containers. Provide the app and its automatically turned into a container and then deployed to Kubernetes |

Memcache -

* Shared - Free and default
* Dedicated - Fixed capacity. Costs extra

--no-promote - Adds app engine application but doesn't send traffic to it. You send it there later slowly.

NoSQL Options -

* Datastore - Cheaper, MBs to TBs
* BigTable - ms response time, MUCH more expensive, TBs to PBs. Not recommended < 1 TB.

---------------------------------------------------------------------------

**Test taking:**

* If you've prepared enough you should finish well under 2 hours. Mallika finished in a little over an hour. I was done with my "first pass" in an hour but spent another 40 minutes trying to get the 16 questions right I wasn't sure about.
* You'll have to give up all the stuff in your pockets before the exam.
* You can go to the bathroom if you need it.
* You can wear a mask but they should socially distance appropriately so you shouldn't have to worry about it.
* When you get to your test taking station draw out the MountKirk and Terrafirm diagrams (from memory).
* You will be provided paper and pencil. Don't be afraid to ask for more paper.
* Jerry's method for answering questions (take it or leave it):
  1. Read each question carefully without looking at answers. Re-read if it doesn't immediately spark recollection.
  2. If you know what the answer should be and can drive to it do so. Then read the answers to make sure there isn't a better option. DO NOT change your answer unless you know you're wrong. Usually the first choice is the right one.
  3. If you don't have the right answer off the top of your head use the elimination method. Write down ABCD on your scratch paper. Read each answer and using the KEYWORDS (above) try to eliminate as many options as possible. Ex. The question says analytical database as one of the requirements. You can immediately cross out any answers that don't have BigQuery in the answer. Actually cross out the invalid options on paper and why you did it. Generally, you'll be left with 2 answers to pick from. You've given yourself a 50% chance to get it right!
  4. If you still can't figure it out mark it for review and move on. Give yourself a break. 5 extremely hard questions in a row can make the test seem harder than it is and destroy your confidence.
  5. ALWAYS select an answer even if you're not sure or mark it for review. If for some reason you submit early you at least have a chance to get it right.

If you pass you'll get a message that says it will take 7-10 days to verify. This is to verify you didn't cheat or register under a different email address to avoid the waiting period between retakes. I assume if you get this you passed. I'm told it also says in the same screen if you look VERY hard. Otherwise you can go to webassesor to find out: <https://www.webassessor.com/home.do?page=CANDIDATE&tabs=1&branding=GOOGLECLOUD>

Special Thanks to Mallika for being the first and bravest among us to take this test and provide key tips like Linux Academy without which I'm not sure I would have passed.