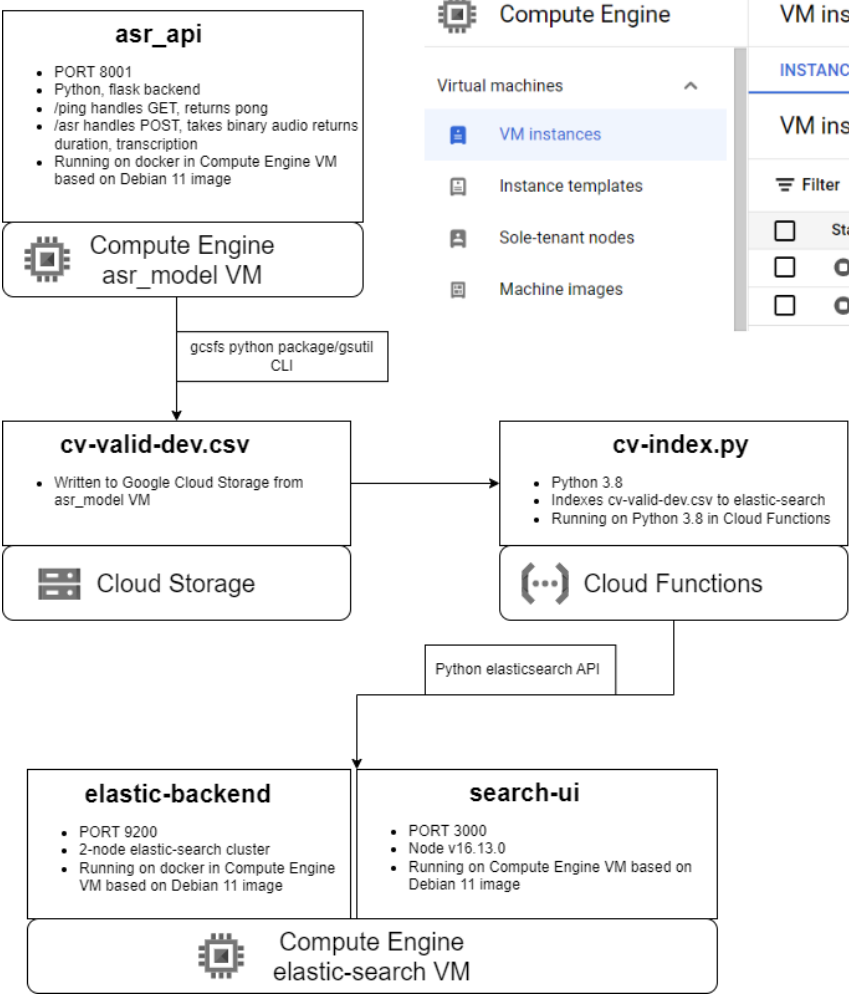


# ASR\_API Architecture



Google Cloud VM instances screenshot showing the VM instances table. The table has columns: Status, Name, Zone, Recommendations, In use by, and Internal IP. The instances listed are **asr-model** and **elastic-search**, both in the **us-west2-a** zone.

Status	Name	Zone	Recommendations	In use by	Internal IP
<input type="checkbox"/>	asr-model	us-west2-a			10.168.0.2 (nic0)
<input type="checkbox"/>	elastic-search	us-west2-a			10.168.0.3 (nic0)

Google Cloud Cloud Storage bucket details screenshot for the **asr-api** bucket. The bucket is located in **us-west2 (Los Angeles)** and has a **Standard** storage class. The bucket contains two objects: **cv-valid-dev.csv** (370.1 KB, text/csv) and **cv-valid-test.zip** (132.8 MB, application/x-zip-compressed).

Name	Size	Type
cv-valid-dev.csv	370.1 KB	text/csv
cv-valid-test.zip	132.8 MB	application/x-zip-compressed

## Additional Notes:

**Compute Engine:** GCP Compute Engine was chosen as a quick and flexible way to deploy a multi-purpose remote server.

In a production environment with higher traffic and scalability requirements, using App Engine or Cloud Run would be more appropriate since they provide a managed Kubernetes backend. In practice, both this CE implementation and AE/CR would both rely on containerization, but some manual setup (such as configuring elastic-search virtual memory) would mean containerization for AE/CR requires more effort

Ideally, search-ui and elastic-backend should be run in separate servers, since elastic-backend is more resource intensive than search-ui and should be independently scalable in a microservice architecture.

**Cloud Storage & Cloud Functions:** Cloud Functions provide broad utility for serverless functions, particularly in a event-trigger context. cv-index.py can technically be run from anywhere, but implementing it in cloud functions along with a pub/sub or event-arc trigger would allow any transcriptions from asr\_api that are saved to cloud storage to be automatically indexed to elastic-backend.

In a production environment, asr\_api would probably be called repeatedly to make transcriptions. Here, using a single .csv in cloud storage would not be ideal and, depending on what is needed, BigQuery could provide a better storage system. This can also be configured with a pub/sub or eventarc trigger to run cv-index.py whenever the BigQuery table is updated with a new transcription.