Python (https://cloud.google.com/python/) Guides

Getting started with Django

Django apps that run on Google Cloud are running on the same infrastructure that powers all of Google's products, which generally improves scalability.

Hosting platforms

There are four main options for deploying Django on Google Cloud:

Django deployment option	Use if you want	Don't use if you need	Get started
App Engine standard environment	• Minimal configuration • No server maintenance • Easy scalability	• Binary executables not available in the App Engine standard Python environment	Django on App Engine standard environment (https://cloud.google.com/python/django/appengine)
App Engine flexible Environment	• Most of the advantages of App Engine • Custom Docker runtimes	• Control over the entire VM (outside of the application's docker container)	Django on App Engine flexible environment (https://cloud.google.com/python/django/managed-vms)
Google Kubernetes Engine (GKE)	• Django containers in a	• A fully featured platform and environment	Django on Google Kubernetes Engine (https://cloud.google.com/python/django/kubernetes-engine)

	microservice environment • A toolkit to design your own container-based platform	that lets developers build apps and services over the internet. For a container-based solution, consider the flexible environment.	
Compute Engine	• Computing infrastructure that uses VMs • Windows VMs	• A serverless environment that doesn't require you to configure your own infrastructure	Django in Google Cloud Marketplace (https://cloud.google.com/marketplace/solution/bitnami-launchpad/djangostack?q=django)

Databases

The Django object-relational mapper (ORM) works best with an SQL relational database. If you are starting a new project, Cloud SQL (https://cloud.google.com/sql/) is a good choice. With a few clicks, you can create a MySQL or PostgreSQL database that's managed and scaled by Google.

You can also use other SQL databases if you're willing to manage them yourself on Compute Engine or another service.

Sometimes, there are compelling reasons to use a NoSQL database, such as scalability or suitability for your data model. Using the Django ORM with a NoSQL database is possible with some limitations: for example, many types of database joins can be expressed in Django, but these joins aren't supported by Datastore and other NoSQL databases like MongoDB.

One possibility is to use a mixed SQL and NoSQL approach that uses different databases for different types of data.

For a managed, massively scalable NoSQL solution, consider Datastore (https://cloud.google.com/datastore/), which is a non-relational database that scales better than a SQL solution.

If you choose to use MongoDB, you can deploy it using Google Cloud Marketplace (https://cloud.google.com/marketplace/solution/click-to-deploy-images/mongodb) and do your own management, or you can use the managed MongoDB hosting service provided by mLab (https://mlab.com/).

For many years, the most popular approach to making the Django ORM work with NoSQL solutions was Django non-rel (https://github.com/django-nonrel/django-nonrel), but the project isn't maintained. Another project, called Djangae (https://github.com/potatolondon/djangae), provides a Django ORM backend for Datastore without forking Django; however, it isn't supported on App Engine.

Caches

App Engine comes with a built-in memcache system (https://cloud.google.com/appengine/docs/python/memcache/usingmemcache?hl=en). To install the memcache system on Compute Engine, use Google Cloud Marketplace (https://cloud.google.com/marketplace/solution/bitnami-launchpad/memcached?q=memcache). To install the memcache system on either Compute Engine or GKE, use the Memcached Docker image (https://hub.docker.com/_/memcached/). Similarly, you can install Redis by using Google Cloud Marketplace (https://cloud.google.com/marketplace/solution/click-to-deploy-images/redis?hl=en) or the Redis Docker image (https://hub.docker.com/_/redis/).

Task queuing

App Engine comes with a built-in task queue feature (https://cloud.google.com/appengine/docs/python/taskqueue/) for long-running background jobs. Outside of App Engine, you can use Pub/Sub (https://cloud.google.com/pubsub/) to queue tasks with Pub/Sub Task Queue for Python (psq) (https://github.com/GoogleCloudPlatform/psq).

Other popular task queuing options that are available in Google Cloud Marketplace include RabbitMQ (https://cloud.google.com/marketplace/solution/click-to-deploy-images/rabbitmq?hl=en) and Kafka (https://cloud.google.com/marketplace/solution/bitnami-launchpad/kafka?q=kafka). There are also Docker images (https://hub.docker.com/) for RabbitMQ and Kafka.

Last updated December 5, 2019.