GettingStartedWithGoogleAppEngine

AppEngine

- **Simplestway** todeployandscaleyourapplicationsin GCP.
- Providesend-to-endapplicationmanagement.
- **Supports**:
- Go,Java,.NET,Node.js,PHP,Python,Rubyusingpre-configuredruntimes.
- Usecustomrun-timeandwritecodeinany language.
- ConnecttovarietyofGoogleCloudstorageproducts (CloudSQLetc).
- **Nousagecharges**-Payforresourcesprovisioned.
- **Features**:
- Automaticloadbalancing&Auto scaling.
- Managedplatformupdates& Applicationhealth monitoring.
- Applicationversioning.
- Trafficsplitting.

ComputeEnginevsAppEngine

* **ComputeEngine**
- IAAS

- MOREFlexibility
- MOREResponsibility
- ChoosingImage.
- InstallingSoftwareChoosingHardware.
- FinegrainedAccess/Permissions(Certificates/Firewalls).
- Avaiability etc.
- **AppEngine**.
- PaaS
- Serverless
- LESSER Responsibility.
- LOWERFlexibility.

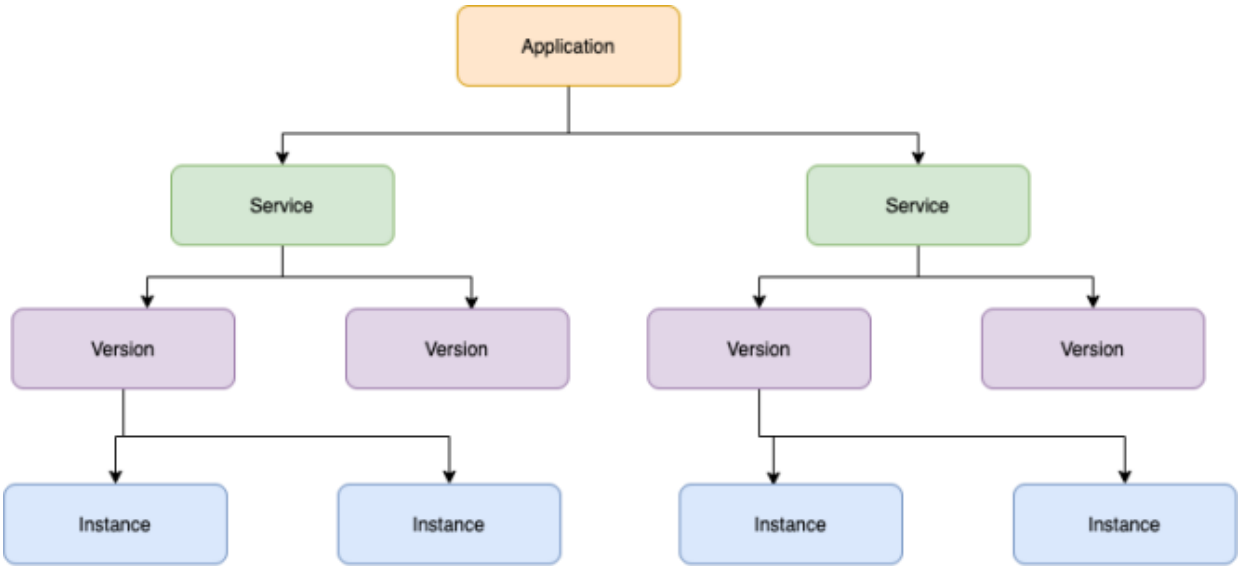
AppEngineenvironments

* **Standard**:Applicationsruninlanguagespecific sandboxes.

- CompleteisolationfromOS/Disk/OtherApps.
- **V1**:Java,Python,PHP,Go(OLD Versions).
- ONLY forPythonandPHPruntimes:
- RestrictednetworkAccess.
- Onlywhite-listedextensionsandlibrariesareallowed.
- NoRestrictionsforJavaandGoruntimes.
- **V2**:Java,Python,PHP,Node.js,Ruby,Go(NEWER Versions).

- FullNetworkAccessandNorestrictionsonLanguage Extensions.
- **Flexible**-ApplicationinstancesrunwithinDocker containers.
- MakesuseofComputeEnginevirtualmachines.
- SupportANYruntime(withbuilt-insupportforPython, Java, Node.js, Go, Ruby, PHP, or .NET).
- Providesaccesstobackgroundprocessesandlocal disks.

AppEngine-ApplicationComponentHierarchy



Application: OneAppperProject.

Service(s): MultipleMicroservicesorAppcomponents.

- Youcanhavemultipleservicesin asingleapplication.
- Each**Service**canhavedifferent settings.
- EarliercalledModules.
- **Version(s):**Eachversionassociatedwithcodeand configuration.
- Each**Version** canruninoneor moreinstances.
- Multipleversionscanco-exist.
- Optionstorollbackandsplittraffic.

Commandsexecutedinnextsteps

1. cddefault-service
2. gcloudappdeploy
3. gcloudappserviceslist
4. gcloudappversionslist
5. gcloudappinstances list
6. gcloudappdeploy--version=v2
7. gcloudappversionslist
8. gcloudappbrowse
9. gcloudappbrowse--version20210215t072907
10. gcloudappdeploy--version=v3--no-promote
11. gcloudappbrowse--version v3
12. gcloudappservicesset-trafficsplit=v3=.5,v2=.5
13. gcloudappservicesset-trafficsplits=v3=.5,v2=.5

14. watchcurlhttps://melodic-furnace-304906.uc.r.appspot.com/
15. gcloudappservicesset-traffic--splits=v3=.5,v2=.5--split-by=random
16. cd../my-first-service/
17. gcloudappdeploy
18. gcloudappbrowse--service=my-first-service
19.
20. gcloudappserviceslist
21. gcloudappregionslist
22.
23. gcloudappbrowse--service=my-first-service--version=20210215t075851
24. gcloudappbrowse--version=v2
25. gcloudappopen-console--version=v2
26. gcloudappversionslist--hide-no-traffic

AppEngine-ScalingInstances

* **Automatic**-Automaticallyscale instancesbasedontheload:
  - RecommendedforContinuouslyRunningWorkloads.
  - Autoscalebasedon:
  - **TargetCPUUtilization**-ConfigureaCPUusagethreshold.
  - **TargetThroughputUtilization**-Configureathroughput threshold.

- **MaxConcurrentRequests**-Configuremaxconcurrent requests an instance can receive.
  - Configure**MaxInstances** and**MinInstances**.
- **Basic**-Instancesarecreatedasandwhen requestsare received:
  - RecommendedforAdhoc Workloads.
  - InstancesareshutdownifZEROrequests.
  - Triestokeepcosts low.
  - Highlatencyispossible.
- NOTsupportedbyAppEngine FlexibleEnvironment.
  - ConfigureMaxInstancesandIdle Timeout.
- **Manual**-Configurespecificnumberofinstancestorun:
  - Adjustnumberofinstancesmanuallyover time.

AppEngineDemo

- Deploy an application to cloud using App Engine.

app.yaml Reference

```yaml
runtime: python28 #The name of the runtime environment that is used by your app
api_version: 1  #RECOMMENDED - Specify here - gcloud app deploy -v [YOUR_VERSION_ID]
instance_class: F1
service: service-name
#env: flex

inbound_services:
- warmup

env_variables:
  ENV_VARIABLE: "value"

handlers:
- url: /
  script: home.app

automatic_scaling:
  target_cpu_utilization: 0.65
  min_instances: 5
  max_instances: 100
  max_concurrent_requests: 50
#basic_scaling:
  #max_instances: 11
  #idle_timeout: 10m
#manual_scaling:
  #instances: 5
```


- First go to google console and create a new project here.
- Now Switch to newly created project.
- Now go to App Engine Admin Api and Enable it in newly created project.
- In another way enabling is go to app engine and select create application.
- Now select the region and kept service account as default.
- Now select python and standard options.
- Now open cloud shell and open editor in itself.
- Now given in course download folder just drag and drop the my first service and default service folders in explorer.
- ls
- cd default service
- gcloud config list
- select authorize
- gcloud config set project <your project ID>

- gcloud app deploy
- Copy the link in output and paste it in browser and check.
- gcloud app services list
- gcloud app versions list
- gcloud app deploy --version=v2
- gcloud app browse list
- gcloud app browse --version <particular version>
- gcloud app deploy --version=v3 –no-promote(command to deploy new version without traffic switching to it)
- gcloud app services set-traffic --splits=v3=.5,v2=.5
- cd ../my-first-service/
- gcloud app deploy
- gcloud app services list
- gcloud app versions list
- gcloud app browse
- gcloud app browse --service=my-first-service --version=<ID>

AppEngine - Request Routing

*You can use a combination of three approaches:
* Routing with URLs:
* https://PROJECT_ID.REGION_ID.r.appspot.com (default service called)
* https://SERVICE-dot-PROJECT_ID.REGION_ID.r.appspot.com (specific service)
* https://VERSION-dot-SERVICE-dot-PROJECT_ID.REGION_ID.r.appspot.com (specific version of service)
* Replace -dot- with . if using custom domain.
* Routing with a dispatch file:

* Configure dispatch.yaml with routes.
* gcloud app deploy dispatch.yaml.
* Routing with Cloud Load Balancing:
* Configure routes on Load Balancing instance.

AppEngine - Deploying new versions without downtime

* How do I go from V1 to V2 without downtime?
* Option 1: I'm very confident - Deploy & shift all traffic at once:
*Deploy and shift all traffic at once from v1 to v2: gcloud app deploy.
* Option 2: I want to manage the migration from v1 to v2.
* STEP 1: Deploy v2 without shifting traffic (--no-promote).
* gcloud app deploy --no-promote.
* STEP 2: Shift traffic to V2:
* Option 1 (All at once Migration): Migrate all at once to v2.
* gcloud app services set-traffic s1 --splits V2=1.
* Option 2 (Gradual Migration): Gradually shift traffic to v2. Add --migrate option.
* Gradual migration is not supported by App Engine Flexible Environment.
* Option 3 (Splitting): Control the pace of migration.
* gcloud app services set-traffic s1 --splits=v2=.5,v1=.5.
* Useful to perform A/B testing.
* Ensure that new instances are warmed up before they receive traffic (app.yaml - inbound_services > warmup)

How do you split traffffic between multiple versions?

- How do you decide which version receives which traffic?
- IP Splitting - Based on request IP address.
- IP addresses can change causing accuracy issues! (I go from my house to a coffee shop).
- If all requests originate from a corporate vpn with single IP, this can cause all requests to go to the same version.
- Cookie Splitting - Based on a cookie (GOOGAPPUID).
- Cookies can be controlled from your application.
- Cookie splitting accurately assign users to versions.
- Random - Do it randomly.
- How to do it?
- Include --split-by option in gcloud app services set-traffic command.
- Value must be one of: cookie, ip, random.
- gcloud app services set-traffic s1 --splits=v2=.5,v1=.5 --splitby=cookie.

Playing with App Engine

- gcloud app browse/create/deploy/describe/open-console
- gcloud app create --region=us-central
- gcloud app deploy app.yaml
- --image-url: Only for flexible environments. Deploy docker image.
- gcloud app deploy --image-url gcr.io/PROJECT-ID/hello-world-rest-api:0.0.1.RELEASE.
- --promote --no-promote (Should new version receive traffic?)

- --stop-previous-version --no-stop-previous-version (Should old version be stopped after new version receives all traffic?)
- --version (Assign a version. Otherwise, a version number is generated.)
- gcloud app browse --service="myService" --version="v1" (open in a web browser).
- gcloud app open-console --service="myService" --version="v1".
- gcloud app open-console --logs.
- Other Commands
- gcloud app logs tail
- gcloud app regions list

Playing with App Engine Instances

- gcloud app instances delete/describe/list/scp/ssh
- gcloud app instances delete i1 --service=s1 --version=v1
- gcloud app instances describe --service=s1 --version=v1 i1
- gcloud app instances list
- gcloud app instances scp --service=s1 --version=v1 --recurse local_dir i1:remote_dir (Copy files to/from App Engine Flexible instances)
- gcloud app instances ssh --service=s1 --version=v1 i1 (SSH into the VM of an App Engine Flexible instance)

Playing with App Engine Services and Versions

* gcloud app services browse/delete/describe/list/set-traffic
* gcloud app services list
* gcloud app services browse myService --version="v1"
* gcloud app services delete service1 service2
* gcloud app services describe service1
* gcloud app services set-traffic APP1 --splits v1=0.9,v2=0.1
* --split_by (ip, cookie, random)
* gcloud app versionsbrowse/delete/describe/list/migrate/start/stop
* gcloud app versions list
* --hide-no-traffic (Only show versions that are receiving traffic)
* gcloud app versions browse/delete/describe v1 --service="myService" .
* gcloud app versions migrate v2 --service="myService" (migrate all traffic to new version).
* gcloud app versions start/stop v1
* --service=my-service Only start v1 of service my-service

App Engine - Cron Job

```
cron:
- description: "daily summary job"
  url: /tasks/summary
  schedule: every 24 hours
```

---

* Allows to run scheduled jobs at pre-defined intervals
* Use cases:
* Send a report by email every day.
* Refresh cache data every 30 minutes.
* Configured using cron.yaml .

* Run this command - gcloud app deploy cron.yaml.
* Performs a HTTP GET request to the configured URL on schedule

Others Important App Engine yaml files

* dispatch.yaml - override routing rules.

```
dispatch:
  - url: "*/mobile/*"
    service: mobile-frontend
  - url: "*/work/*"
    service: static-backend
```

* queue.yaml - manage task queues.

```
queue:
- name: fooqueue
  rate: 1/s
  retry_parameters:
    task_retry_limit: 7
    task_age_limit: 2d
```

App Engine – Remember

- AppEngine is Regional (services deployed across zones).
- You CANNOT change an Application's region.
- Good option for simple microservices (multiple services).
- Use Standard v2 when you are using supported languages.
- Use Flexible if you are building containerized apps.
- Be aware - ATLEAST one container is always running when using .

- Flexible: Go for Standard if you want to be able to scale down the number of instances to zero when there is NO load.
- Use a combination of resident and dynamic instances.
- Resident Instances: Run continuously.
- Dynamic Instances: Added based on load.
- Use all dynamic instances if you are cost sensitive.
- If you are not very cost sensitive, keep a set of resident instances running always.

App Engine – Scenarios

Scenario	Solution
I want to create two Google App Engine Apps in the same project	Not possible. You can only have one App Engine App per project. However you can have multiple services and multiple version for each service.
I want to create two Google App Engine Services inside the same App	Yup. You can create multiple services under the same app. Each service can have multiple versions as well.
I want to move my Google App Engine App to a different region	App Engine App is region specific. You CANNOT move it to different region. Create a new project and create new app engine app in the new region.
Perform Canary deployments	Deploy v2 without shifting traffic (`gcloud app deploy --no-promote`) Shift some traffic to V2 (`gcloud app services set-traffic s1 --splits v1=0.9,v2=0.1`)