App Engine

App Engine 92 Compute Engine IAAS MORE Flexibility MORE Responsibilit y C h o o sing Im a g e Installing Softwa r e C h o o sing H a r d w a r e Fin e gr ain e d A c c e s s / P e r mis sio n s ( C e r tific a t e s / Fir e w a l l s ) A v ail a bilit y e t c App Engine PaaS Serverless LE S S E R Res ponsibilit y L O W E R F l e xibilit y C o mpu t e E n g i n e v s App E n g i n e 93 Standard: Applications run in language specific sandboxes Complete isolation from OS/Disk/Other Apps V1: Java, Python, PHP, Go (OLD Versions) ONLY for Python and PHP runtimes: Restricted network Access Only white-listed extensions and libraries are allowed No Restrictions for Java and Go runtimes V2: Java, Python, PHP, Node.js, Ruby, Go (NEWER Versions) Full Network Access and No restrictions on Language Extensions Flexible - Application instances run within Docker containers Makes use of Compute Engine virtual machines Support ANY runtime (with built-in support for Python, Java, Node.js, Go, Ruby, PHP, or .NET) Provides access to background processes and local disks App Engine environments 94 Application: One App per Project Service(s): Multiple Microservices or App components You can have multiple services in a single application Each Service can have different settings Earlier called Modules Version(s): Each version associated with code and configuration Each Version can run in one or more instances Multiple versions can co-exist Options to rollback and split traffic App Engine - Application Component Hierarchy 95 App Engine - Comparison Feature Standard Flexible Pricing Factors Instance hours vCPU, Memory & Persistent Disks Scaling Manual, Basic, Automatic Manual, Automatic Scaling to zero Yes No. Minimum one instance Instance startup time Seconds Minutes Rapid Scaling Yes No Max. request timeout 1 to 10 minutes 60 minutes Local disk Mostly(except for Python, PHP). Can write to /tmp. Yes. Ephemeral. New Disk on startup. SSH for debugging No Yes 96 Automatic - Automatically scale instances based on the load: Recommended for Continuously Running Workloads Auto scale based on: Target CPU Utilization - Configure a CPU usage threshold. Target Throughput Utilization - Configure a throughput threshold Max Concurrent Requests - Configure max concurrent requests an instance can receive Configure Max Instances and Min Instances Basic - Instances are created as and when requests are received: Recommended for Adhoc Workloads Instances are shutdown if ZERO requests Tries to keep costs low High latency is possible NOT supported by App Engine Flexible Environment Configure Max Instances and Idle Timeout Manual - Configure specific number of instances to run: Adjust number of instances manually over time App Engine - Scaling Instances 97 AppEngine Demo Deploy an application to cloud using App Engine 98 app.yaml Reference 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 99 You can use a combination of three approaches: Routing with URLs: (default service called) (specific service) (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 - Request Routing https://PROJECT\_ID.REGION\_ID.r.appspot.com https://SERVICE-dot-PROJECT\_ID.REGION\_ID.r.appspot.com https://VERSION-dot-SERVICE-dot-PROJECT\_ID.REGION\_ID.r.appspot.com 100 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) AppEngine - Deploying new versions without downtime 101 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 How do you split traffffic between multiple versions? 102 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 103 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 Instances 104 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 versions browse/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 105 App Engine - Cron Job 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 cron: - description: "daily summary job" url: /tasks/summary schedule: every 24 hours 106 Others Important App Engine yaml files dispatch.yaml - override routing rules queue.yaml - manage task queues dispatch: - url: "\*/mobile/\*" service: mobile-frontend - url: "\*/work/\*" service: static-backend queue: - name: fooqueue rate: 1/s retry\_parameters: task\_retry\_limit: 7 task\_age\_limit: 2d 107 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 - Remember 108 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 --nopromote) Shift some traffic to V2 (gcloud app services set-traffic s1 --s