Tech Challenge-1: Logging and Monitoring

As per GCP best practices, a dedicated GCP project for Logging and Monitoring is recommended (may not be possible in your sandbox environment)

More requirements are as follows:

1. For Tech Challenge 2, you are required to store Cluster, Pods and Container creation Logs for an indefinite period in a GCP service offering your review team a SQL interface. All logs should be streamed in real time in this GCP service.
2. For Tech Challenge 3, all Compute Engine logs may never be used but need to be archived for ever (think cost).
3. Create a combined dashboard with a Graphical User Interface for all projects of Tech Challenge 2 and 3 to monitor all consumed GCP resources (storage, compute, Network etc.)
4. Use Metrics Explorer to plot “The total number of log entries that were exported using sinks” and also “The total number of bytes exported using sinks”

Hint - IAM and Stackdriver

Tech-Challenge-2: Google Kubernetes Engine Deployment

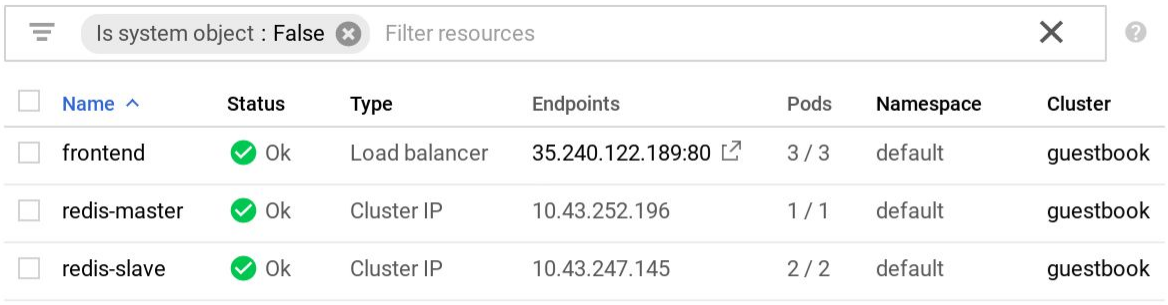
The objective of this challenge is to explore 3 Kubernetes deployment configurations through a very simple 2 tier web application.

For this challenge we will use the Guestbook Application.

3 deployment configurations are as follows:

1. Case 1: Implement the frontend and backend service on same GKE Cluster on one single VPC (basically same as shown in the tutorial) - remember to convert the frontend service to type load balancer

Desired State of Services will be as shown below



2. Case 2: Implement backend and frontend services on different GKE Clusters on the same VPC. The frontend should access the backend services through a VPC internal IP address. Backend services should communicate with each other using VPC internal IP address

   For this case you can copy the necessary yaml files from
   gs://tech-challenge-solutions/Tech-Challenge-2-Case-2

   A good read: Kubernetes Options

   Create two new zonal kubernetes clusters

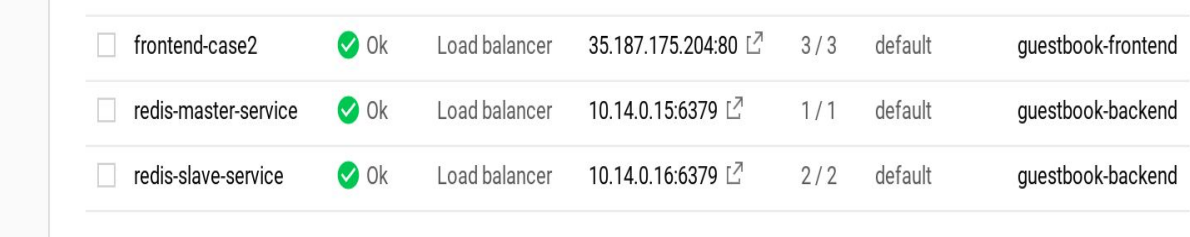
   A. We need to have the backend redis services with type internal load balancer to achieve this.
   B. Redis Master Deployment has no change from Case 1 (you can continue using the same deployment), Redis Master Service will change (with type Internal Load Balancer) - Note the IP address of Redis Master Service.
   C. Redis Slave Deployment needs to be modified to have the IP address of the Redis Master Service created in the previous step. Redis Slave Service will change (with type Internal Load Balancer) - Note the IP address of Redis Slave Service
   D. Both the Service IP addresses will now be a private IP address and not a cluster IP address.

Create a new Kubernetes Cluster for Frontend Deployment and Services

   A. Create a new Frontend Deployment, it will contain the IP addresses of the Redis Master and Redis Slave Services it accesses.
   B. Create a new Front end Service to test the application - you should see the same performance/features as for Case 1

All necessary YAML files are put in the bucket mentioned above. **Make sure you use the correct ip addresses from your deployment/project**

Desired state of services is like this (note the internal and external IP addresses)



frontend-case2	Ok	Load balancer	35.187.175.204:80	3 / 3	default	guestbook-frontend
redis-master-service	Ok	Load balancer	10.14.0.15:6379	1 / 1	default	guestbook-backend
redis-slave-service	Ok	Load balancer	10.14.0.16:6379	2 / 2	default	guestbook-backend

3. Case 3: Implement backend and frontend services on different VPCs (in different regions). The two VPCs should be connected through a Cloud VPN. The frontend should access the and backend services through an internal IP address.

   Use Classic VPN for this lab to keep things simple.

   A. Setup two projects one for backend pods/services and other for frontend pods/services (use the project of your neighbour as you are not allowed to create another project)
   B. Use the same deployment methodology as case 2 (backend redis pods exposing services through an ILB, redis slave deployment carrying the service ip of the redis master)
   C. Frontend deployment carrying service IP addresses of the redis master and slave
   D. Setup a VPN between the two VPCs (Front end Service will access the redis master and slave services over the VPN tunnel)
   E. VPN tunnels should have Route Based Routing to add Static Routes to GKE Node, Cluster and Services CiDR ranges

   Hint Use `gcloud container clusters describe <cluster-name> --region=<region-name> | grep Ipv4Cidr`

   F. Check if you have the necessary firewall rules open (port 6379, protocol tcp)
   G. Test the system.

Tech-Challenge-3: Infrastructure Deployment

The objective of this challenge is to get acquainted to Deployment Manager (a handy tool for enterprise grade automated deployments), Instance Templates, L7 Load Balancer, Virtual Machine and Startup scripts and Metadata server.

Perform the following:

1. Using Deployment Manager or Cloud Console (combination of python/jinja templates and configuration YAML file)

a) Create an empty Custom VPC
b) Create 2 Subnet in any of the regions
c) Create Firewalls (to allow HTTP and SSH traffic from Internet - priority 786, allowing all internal subnet traffic - priority 1000)
d) Do you know that in Cloud Deployment Manager you can preview a configuration prior to deployment (look for --preview flag)

If you want to try deployment manager, the prepared deployment files are in gs://tech-challenge-solutions. Copy them in a local folder and run the following command

gsutil cp gs://tech-challenge-solutions/Tech-Challenge-3-Deployment-Manager/* .

Explore the yaml main config file and jinja files for network, subnetwork and firewall creation. Jinja files are based on Network API Subnetwork API Firewall API - notice the similarity between the template and the API structure. If possible try to change parameters to get a better grasp of the jinja file.

Create a deployment using the deployment manager gcloud command.

```
gcloud deployment-manager deployments create test-deployment --config config.yaml
```

Verify following message in cloud shell

```
Create operation operation-1569489296795-593713403a904-9b79a41c-8292af2d completed successfully.
NAME                    TYPE                      STATE      ERRORS INTENT
fw-allow-all-internal   compute.v1.firewall       COMPLETED  []
fw-allow-http-ssh       compute.v1.firewall       COMPLETED  []
sn-0-us-central1        compute.v1.subnetwork     COMPLETED  []
sn-1-europe-west2       compute.v1.subnetwork     COMPLETED  []
xxx-yyy                 compute.v1.network        COMPLETED  []
```

Also verify network, subnetwork and firewall rules are created in the cloud console.

For further steps use the Cloud Console.

2. Create an Instance Template for VMs to be deployed in created subnets. Select Instance Template under Compute Engine, select an appropriate name, keep default values, for firewalls select allow HTTP and HTTPS traffic. **Select machine type f1-micro to optimize costs and easily test load balancing later.**

In Metadata, add following key value pairs.
startup-script-url (key)
gs://tech-challenge-solutions/Tech-Challenge-3-StartupScript/startup-script-v1.sh (value)

Feel free to explore the startup script for installation of Apache web server. (Startup Script queries the compute engine metadata server for collecting information about the instance)

3. Create a test-vm based on the created instance template in step 2. Click on the created instance template and on top you will see an option to create VM. In regions select regions for one of the subnets you created earlier and click on create at the bottom of the page. On the VM instances page see the created VM, note the public IP address and do **http://<public-ip>** you should see an output like this

Hi Some Information About Me - I am Version 1

My Name is : inst-temp-1

My External IP is : 34.93.8.255

My Internal IP is : 10.20.0.4

After verifying delete the VM
.

4. Create a Regional Managed Instance Group - give it a suitable name, select the other region in which you have your subnet. Put **max number of instances to 9, minimum is 3**. In Auto Scaling parameters set **CPU Utilization to 20%**, set max instances to 7. Create a Health Check (set value for check_interval to 90 sec and timeout to 15 seconds) for autohealing of Unhealthy VMs.

6. You should see 3 VMs listed in VM instances page with names starting with the name of your managed instance group.

The created VMs should display the following information on its index homepage (simply http://<ip-address>

a) Name
b) External IP
c) Internal IP

7. Configure a backend service on a L7 load balancer - for Loadbalancer to communicate with the Backend services you need to create firewall rules - Read this page to understand the concept. (I have not told you what exactly needs to be done so figure it out because without it further steps will not make sense:-))

8. Explore the Load balancer and make sure that backend is being reported healthy.

Load balancer can take up to 10 minutes to provision.
You can test Load Balancer Status using
ILB_IP=<load-balancer-ip>
for ((i=1;i<=100;)); do curl $ILB_IP; sleep 3; done

4. Create a small VM to test your Load Balancer and see the impact of auto scaling
Machine-type fi-micro with default debian image
ssh into the machine
Install Apache Server Benchmarking Tool

```
sudo apt-get install apache2-utils
```

Stress Test Using this simple 1 liner

```
while true; do ab -kc 400 -n 5000 -t 600
  http://<Global-LB-IP-Address>/index.html; sleep 1; done
```

4. Use Apache Load Testing to load test your load balancer (see impact in cloud console)
5. Try Changing the auto scaling parameters of instance groups, removing a group from the backend service to see the traffic patterns in the load balancer.

6. Create a new instance template, same as the earlier one you created except change the startup script to following value - we are introducing a small change in the startup script.
startup-script-url (key)
gs://tech-challenge-solutions/Tech-Challenge-3-StartupScript/startup-script-v2.sh (value)

7. Explore the rolling update feature of Google Compute Engine for canary testing in live environments. In the new VMs you will see a VMs to display VM Zone and Network name as well.