Transverse monitoring solution for GCP resources

- Owners:
  - *Ahmed Belhoula* @ahmed.belhoula
- Status:
  -

  -

- Scope:
  - Compute engine
  - GKE
  - Cloud SQL
  - Cloud Functions
  - Perf metrics

Introduction

SLA and SLO are closely related but have slightly different purposes. SLA provides a legally binding agreement for service availability and performance, while SLO provides a target level of service availability and performance that the customer can expect. The SLO is used to monitor and measure the performance of GCP services, while the SLA is used to provide a guarantee of service availability and performance.

Goals

- Define Basic metrics for used gcp services
- Alert/mangement automation based on metrics alert threshold with terraform
- Deploy custom Dashboards for used gcp services
- Test Infra Terraform manifest

Proposal

Build Alerting stack with gcp native services

Metrics

The list of the metrics identified base on the gcp performance Metrics scope:

- CPU utilization on compute engine services (CoS & GCE & cloud VMs)
- Memory utilization
- Disk utilization
- Requests per minute
- The error rate

SLO/SLAs

Describe slo (service level objectif) here based on yout sp (service providers) And With those SLO you can compare with the SLAs for your sc(service consumers) Native GCP monitoring stack for gke,compute engine and cloud sql

Define SLOs based on business model/critcatly & metrics

Example for SaaS applications critcatly levels :

- Strategic : Big projects/Clients applications and serivces
- Critcal : Critical app & services
- Standard : reset of applications

Bootsrap

- provision/dev:

  - folder `dev` : Manifsets for creating zonal gke cluster with seperate node pool (2 nodes) CoS optimized and seprate vpc/subnet in europe-west-9 and a cloud function that runing small python script
  - monitoring folder contains all manifset to deploy the monitoring/alerting solution.

- modules:

  - Sa-admin: admin service account for platform admin team memeber
  - network: vpc and subnet setup
  - GKE : Creating gke cluster with sperate node pool

- Backend:

  - GCS PAth : cantech-terraform-tf/monitoring

```
cd provision/TF/dev
terraform init
terraform plan
terraform apply
```

- VM WITH agent

```
cd provision/TF/dev/monitoring/vm-agent
terraform init -var project=cantech-terraform
terraform plan -var project=cantech-terraform
terraform apply -var project=cantech-terraform
```

Deploy dashboards with tf

```
cd provision/TF/dev/monitoring/
export GOOGLE_APPLICATION_CREDENTIALS=/home/ahmed/workspace/hack/gcp-monitoring/provision/TF/dev/gcp-keys/cantech-terraform-gcp-admi

terraform apply -var 'dashboard_json_file=dashboards/gke-compute-resources-cluster-view.json'  -var 'project_id=cantech-terraform'

terraform apply -var 'dashboard_json_file=dashboards/gce-vm-instance-monitoring.json'  -var 'project_id=cantech-terraform'
```

Links

- GKE Compute Resources - Cluster View dashboard (https://console.cloud.google.com/monitoring/dashboards/builder/2fd8e566-a055-4030-ae4c-d95034d90afe;duration=PT1H;tzId=Europe%2FParis?project=cantech-terraform)

- GCE VM Instance Monitoring (https://console.cloud.google.com/monitoring/dashboards/builder/b0b0956b-87ba-4d68-9954-ded77a61f595;duration=PT1H;tzId=Europe%2FParis?project=cantech-terraform)

- List of alerts (https://console.cloud.google.com/monitoring/alerting?project=cantech-terraform)

- Parc VMs view (https://console.cloud.google.com/monitoring/dashboards/resourceList/gce_instance?project=cantech-terraform&startTime=20230122T202216%2B01:00&endTime=20230122T212216%2B01:00)