GCP: Logging, Monitoring, Observability
2023 Ivan Vlad S.

❶ Monitoring in GCP

- Overview
  - Capture signals
    - Metrics
      - Apps, services, platform, microservices
    - Logs
      - apps, services, platform
    - Trace
      - apps - analyze latency
  - Visualize and analyze
    - Dashboards
    - Metrics Explorer
    - Logs Explorer
      - (used to be Logs Viewer)
    - Service Monitoring
    - Health Checks
    - Debugger
    - Profiler
  - manage incidents
    - Alerts
    - Error Reporting
    - SLO
  - troubleshoot
- Ops-based tools
  - monitoring starts with Signal data

    - logging is all about
        - collect
            - > automatic logging on all app engine, cloud run, GKE and compute engine VMs
        - analyze
            - > analyze log data in real time with logs explorer, pub/sub, dataflow, and bigquery
              > analyze archived logs from cloud storage
        - export
            - > export to cloud storage, or pub/sub, or bigquery
              > export logs-based metrics to monitoring
        - retain
            - > data access logs are retained for 1-3650 days, admin logs for 400 days
              > longer retention available in cloud storage or big query
    - available logs
        - cloud audit logs
            - who did what, where, admin activity, data access, system event, access transparency
        - agent logs
            - fluentd agent, common 3P apps, system software
        - network logs
            - VPC flow, firewall rules, NAT gateway, load balancer
    - error reporting
    - service monitoring
        - understand and troubleshoot intra-service dependencies
- app performance mgmt tools
    - debugger
        - real-time app debugging
        - increased collab by sharing debug sessions
        - debug snapshots, logpoints, conditional debugging
    - trace

        - distributed latency analysis
        - near real-time
        - find performance degradations in apps
    - profiler
        - improve performance and reduce costs
        - understand your apps call patterns
        - low-impact production CPU and heap profiling

2 Avoiding Customer Pain

- Why monitor?
    - monitoring reveals urgent attention, trends, planning, improvements
    - provides continual improvement, dashboards, alerting, debugging
    - Set proper expectations
    - monitoring systems should address what is broken and why
        - symptom and cause
- critical measures
    - metrics help measure success
        - business
            - ROI
            - earnings before interest and taxes (EBIT)
            - employer turnover
            - customer churn
        - software
            - pageviews
            - user registrations
            - click-throughs
            - checkouts
    - metrics should be SMART
        - specific
        - measureable

        - achieveable
        - relevant
        - timebound
    - the 4 golden signals
        - latency: impacts user experience
            - > indicate emerging issues
              > may be tied to capacity demands
              > may be used to show improvements
        - traffic: indicates current system demand
            - > historical trends are used for capacity planning
              > core to calculating infra spend
        - saturation: how full the service is
            - > focuses on most constrained resources
              > frequently tied to degrading performance
        - errors: indicates something is failing
            - > may in indicate config or capacity issues
              > can indicate SLO violation
              > time to alert?
- SLIs, SLOs, SLAs
    - SLI = service level indicator (things you measure)
        - quantifiable measure of service availability
    - SLO = service level objective (an achievable target)
        - a reliability target for an SLI
        - a principled way to agree on the desired reliability of a service
    - services need SLOs
        - customer happiness test :: happy = meet SLO, not happy = missed SLO
        - error budgets
            - an SLO implies an acceptable level of unreliability, this is a budget that can be allocated
            - spend on new feature releases and expected system changes, planned downtime, hardware failure, risky experiments
    - ⭐ most important feature of any system is its reliability

- choosing a good SLI
  - needs to be things we can measure that correlate to the happiness of our users
    - SLI formula: (good events/valid events) x 1000%
      - 3-5 SLIs
- specifying SLIs
  - request/response
    - availability, latency, quality
  - data processing
    - coverage, correctness, freshness, throughput
  - storage
    - throughput, latency
- developing SLOs and SLIs
  - what performance does the business need?
  - user expectations are strongly tied to past performance
    - set SLOS based on past performance and business needs

3 Alerting Policies

- Developing an alerting strategy
  - Alert = automated notification sent by GCP through some notification channel to an external app, ticketing system or human
    - alerts are based on events in a time series
  - Goal: human gets notified when needed
    - a service is down, SLOs or SLAs are not being met, something needs to change
      - e.g., When error budget in danger: Alert!
  - Evaluating alerts
    - Precision (measure of exactness)
    - Recall (measure of completeness)
    - when error count > budget = Alert!
  - window length
    - window = regular length subdivision of the SLO in total time

- use small windows
  - faster alert detection, shorter reset time, poor precision
- use longer windows
  - better precision, reset and detection times longer, spend more error budget before alert
- add a duration for better precision
  - use multiple conditions for better precision and recall
- prioritize alerts based on customer impact and SLA

- Creating alerts
  - defined using alert policies
    - alert policy has a name, one or more conditions, notifications, documentation
      - conditions: what's watched and when to alert
      - can control notification channels
    - use multiple criteria to create resource groups
      - monitor all resources in a group together
- Creating alerting policies with the CLI
  - both CLI and API require alert policy be defined in JSON file
  - gcloud and the API can create, retrieve, and delete alerting policies
- Service Monitoring
  - helps with SLO and alert creation
    - consolidated services overview
      - error budget details
  - access through GCP console or Service Monitoring API
    - select latency or available metrics to act as SLIs
      - compliance periods - set compliance periods, type, and goal
      - configure alert condition for SLO burn rate
    - use SLIs to easily create SLOs
      - windows-based vs request-based SLOs
    - alerting easily integrated

- Monitoring critical systems
  - monitoring is configured via Workspaces
    - single pane of glass, cross project visibility, monitor resources in GCP and AWS
      - centralize and consolidate resource monitoring
    - a workspace belongs to a single host project
      - one workspace can monitor multiple projects
      - multiple workspaces can limit access
      - monitor by project for max isolation
    - IAM roles control user access to workspace
      - monitoring viewer, editor, admin
      - services may. need perms to add metric data: monitoring metric writer
- Understanding dashboards
  - view and analyze metrics
    - predefined dashboards
      - dashboards broken into charts
- Creating charts
  - start with Metrics Explorer
    - view data that you don't need to display long term on a dashboard
- uptime checks
  - check public service availability
    - what makes a good uptime check?
      - > protocol, host, and port are appropriate
        > response checked for specific content

4 Configuring GCP services for observability

- Monitoring
  - OS monitoring agent
    - gathers system and application metrics from VM instances and sends them to monitoring
    - install docs on google site
- logging

        - OS logging agent
            - streams logs from common 3P apps and system software to GCP logging
            - install docs on google site
    - Baking an image
        - goal = org treat the image creation process as a standard DevOps pipeline: > commits to a code base trigger, build jobs, which create, test and deploy images with all requisite software and apps built in, including the logging and monitoring agents
            - Base OS install/GCE public image >> hardened OS image >> platform image >> app image
        - hashicorp Packer can automate image builds, integrates well with GCP
    - non-vm resources
        - AppEngine
            - standard and flex support logging and monitoring
                - logs viewable under GAE app resource
        - GKE Monitoring & Logging
            - K8s Engine Dashboard
        - Prometheus
            - optional monitoring tool for K8s
                - install prometheus and the collector
            - service metrics using Prometheus exposition format can be exported and made visible as external metrics

5 Monitoring Network Security and Audit logs

- Network Security & Audit Logs
    - VPC flow logs > part of Andromeda
        - record a sample of network flows
            - record about one out of every 10 packets of network flows sent from and received by the VM instances, including K8s engine nodes
        - enable VPC flow logs per VPC subnet
            - analyze logs in BQ and visualize in Data Studio
    - VPC firewalls
        - firewall rules logging

            - enable firewall rule logging in the console
            - provide micro-segmentation
        - troubleshooting: using rules to catch incorrect traffic
    - Cloud NAT logs
        - Cloud NAT allows GCE VMs with no external IP to send and receive packets via the internet
            - fully managed service, software defined, grounded in Andromeda
        - logging allows you to log NAT connections and/or error - TCP and UDP only
            - view filtering logs in Logs Explorer
    - Packet mirroring
        - visualize and protect your network, clones VPC instance traffic and FWDs for examination
            - happens at NIC, not part of VPC
    - network intelligence center
        - centralized network monitoring visibility
            - > Topology: view VPC topology and metrics
              > Connectivity tests: prevent outages
              > Performance dashboard: packet loss metrics aggregated across zones
              > Firewall insights: metrics help understand and optimize firewall configs
    - Audit logs
        - who did what, where, and when?
          >> admin activity >> data access >> system event
            - data access logs need to be enabled, not enabled by default

6 Investigating App Performance Issues

- Debugger
    - inspect state of a running app in real-time without stopping or slowing it down
        - debugger must be enabled
        - dynamically add log messages with Log Points
- Trace

- Cloud Trace tracks app latency
  - trace = collection of spans, span = object that wraps metrics and other contextual info about a unit of work in your app

- Profiler
  - statistical, low-overhead memory and CPU profiler > understand performance
    - CPU time, Heap, allocated heap, contention, threads, Wall time (how long it takes to run a block of code)
      - Subtopic 1