Case Studies

Case Studies

In 28 Minutes

- Case Studies are **very important** for the certification exam
  - Download - ***https://cloud.google.com/certification/guides/professional-cloud-architect***
  - You will be able to read the case study during the exam:
    - BUT **I would NOT recommend** depending on it
    - Have a good overview of the case studies before you go to the exam
- **Best Attitude:**
  - **Test yourselves as an architect** using the case studies:
    - Knowledge of Google Cloud Services
    - Exploration Skills: ***https://cloud.google.com/architecture***
    - Understanding Problem Domain: AI, ML, Gaming, IOT, Healthcare, Live Streaming
  - Download the case studies and spend sometime with them
    - Do your analysis and form your opinions
    - Do NOT treat my recommendations as final! (Think and challenge them!)
- **Exam Tip:**
  - **Try to group questions** for each case study and answer them together

Case Study - EHR Healthcare - Overview

In28 Minutes

- Provides **SaaS based electronic health record software** for multi-national medical offices, hospitals, and insurance providers (Exponential growth):
- **Current Architecture:**
  - Multiple colocation facilities with similar but separate environments (lease on one data center is about to expire)
    - **Web-based customer-facing applications** running on Kubernetes clusters
      - Requirement: Consistent way to manage apps that are container-based
      - Requirement: Maintain and manage multiple container-based environments
    - **Databases**: MySQL, MS SQL Server, Redis, and MongoDB
    - **Legacy file- and API-based integrations** with insurance providers
      - Scheduled for replacement over next few years - Will NOT be moved to Google Cloud
      - Requirement: Maintain legacy interfaces to insurance providers (connectivity to on-premises and cloud)
    - Users are managed via Microsoft Active Directory
    - Monitoring with various open source tools
    - Alert by email (generally ignored)

Case Study - EHR Healthcare - Requirements

In 28 Minutes

- **Google Cloud will replace** their current colocation facilities
- **Scalable, resilient platform** spanning multiple environments seamlessly
- **Facing outages** with mis-configured systems, inadequate capacity to manage spikes and inconsistent monitoring practices
- **Other Requirements**:
  - Dynamically scale & provision environments, adapt DRP, continuous deployment
    - 99.9% availability, Reduced latency, regulatory compliance, Decrease infrastructure administration costs
  - Centralized visibility and proactive action on system performance and usage
    - Provide consistent logging, log retention, monitoring, and alerting capabilities
  - On-board new insurance providers quickly
    - Create interfaces to ingest and process data from new providers
  - Insights and predictions:
    - Make predictions and generate reports on industry trends based on provider data
  - Secure and high-performance connection between on-premises systems and Google Cloud

Case Study - EHR Healthcare - Discussion

In 28 Minutes

Service	Discussion
Anthos	Run Kubernetes clusters anywhere (cloud and on-premises) Config Mgmt - Central policies - Kubernetes API, Service Mesh, Access control Service Mesh (dashboards, logging, monitoring, distributed tracing) CI/CD - Watch for updates in the Git repository and applies changes to all relevant clusters automatically
Cloud Logging, Cloud Monitoring	Alerting Policies for Notifications
Cloud Logging > (Cloud Storage, BigQuery)	Log Retention
BigQuery	Make predictions and generate reports on industry trends Batch: Cloud Storage > Dataflow > (BigQuery, ...) Stream: Cloud Pub/Sub > Dataflow > (BigQuery, ...)

Case Study - EHR Healthcare - Discussion - 2

In28 Minutes

Service	Discussion
Cloud Dedicated Interconnect	Secure and high-performance connection
Cloud CDN	Reduced latency
Active Directory Federation Services (AD FS)	For single sign-on Google Cloud Directory Sync (synchronize users and groups from Active Directory to Cloud Identity)
Databases	MySQL, MS SQL Server => Cloud SQL Redis => Memorystore MongoDB => (Deploy using Cloud Marketplace or Use Datastore)

Case Study - Helicopter Racing League - Overview

In 28 Minutes

- **Global sports league** for competitive helicopter racing
  - World championship and several regional league competitions every year
  - Offers paid streaming service for races all over the world with live telemetry and predictions
- **Current Architecture:**
  - **Existing public cloud provider** (public cloud-first company) hosts:
    - Core of mission-critical apps
    - Existing content is stored in an object storage service
    - Race predictions are performed using TensorFlow running on VMs
      - Allows predicting race outcomes but lacks the facility to:
        - Support real-time predictions during races (Requirement)
        - Capacity to process season-long results (Requirement)
  - Video recording and editing is performed at the race tracks:
    - Content is encoded and transcoded in the cloud (using VMs created for each job)
  - Enterprise-grade connectivity and local compute provided by truck-mounted mobile data centers

Case Study - Helicopter Racing League - Requirements

In28 Minutes

- **Migrate existing service to a new platform**
  - **Expand use of managed AI and ML services** to facilitate race predictions:
    - Increase predictive capabilities during and before races:
      - Race results, Mechanical failures, Crowd sentiment
      - Enhanced video streams that include predictions of events within the race (e.g., overtaking)
      - Maintain or increase prediction throughput and accuracy
    - Expose the predictive models to partners
    - Create real-time analytics of viewer consumption patterns and engagement
      - Increase telemetry and create additional insights
      - Measure fan engagement with new predictions
  - **Move the serving of real-time and recorded content** closer to their users:
    - Reduce latency, enhance global availability and quality of the broadcasts
    - Increase the number of concurrent viewers
- **Create a data mart** to enable processing of large volumes of race data
- Increase transcoding performance
- Create a merchandising revenue stream

Case Study - Helicopter Racing League - Discussion - 1

In 28 Minutes

Service	Discussion
BigQuery + AI Platform	Migrate existing service to a new platform (Managed AI and ML services using TensorFlow) Create a data mart to enable processing of large volumes of race data (season-long results) Consider BigQuery ML for realtime predictions
Cloud Storage, Cloud CDN	**Move the serving of real-time and recorded content** closer to their users
Apigee	Expose the predictive models to partners
Cloud Pub/Sub > Cloud Dataflow > Cloud BigQuery	Create real-time analytics of viewer consumption patterns and engagement
Transcoder API	Increase transcoding performance Beta - process from/to Cloud Storage

Case Study - Helicopter Racing League - Discussion - 2

In 28 Minutes

Service	Discussion
Video Intelligence API **Streaming API**	Real-time streaming analysis for live media AIStreamer ingestion library allows you to connect to API Supports File Streaming, HTTP Live Streaming (HLS), Real Time Streaming Protocol (RTSP), Real Time Messaging Protocol (RTMP)
References	***https://cloud.google.com/architecture/serving-machine-learning-models-using-apigee-edge-and-ml-engine*** ***https://cloud.google.com/architecture/mobile-gaming-analysis-telemetry***

Case Study - Mountkirk Games - Overview

In28 Minutes

- **Online, session-based, multi-player games** for mobile platforms
  - Longterm Requirement: Support multiple gaming platforms (expand beyond mobile)
- **Current Architecture**:
  - Migrated on-premises environments to Google Cloud
  - Used Lift-and-shift VM migration for 5 Games
  - Isolated Google Cloud project for each new game:
    - Legacy games with low traffic consolidated to one project
    - Permissions and network policies managed at folder level
  - Separate environments for development and testing

Case Study - Mountkirk Games - Requirements

In28 Minutes

- Working on **retro-style first-person shooter (FPS)** game
  - **Hundreds of simultaneous players** joining a **geo-specific digital arena** from multiple platforms & locations
  - **Proposed**: Run game backend on Google Kubernetes Engine (Rapid Scaling)
    - Google's global load balancer to route to closest regional game arenas
    - Support eventual migration of legacy games to this new platform
    - Requirement: Real-time global leaderboard of top players across every active arena
      - Proposed: Use multi-region Spanner cluster for near real–time global leader board
- **Other Requirements:**
  - Store game activity logs in structured files for future analysis
  - Use GPU processing to render graphics server-side for multi-platform support
  - Enable advanced analytics capabilities
  - Use cloud-native design principles
    - Multiple regions, Rapid iteration, Minimize latency and costs, Auto scaling
    - Managed services and Pooled resources

Case Study - Mountkirk Games - Discussion - 1

In28 Minutes

Service	Discussion
Game Servers (Agones + Kubernetes)	Deliver seamless multiplayer gaming experiences
MIGs + VMs + Global Load Balancing + GPUs	If you need a lot of customization
Cloud Storage	Store game activity logs in structured files for future analysis
Cloud Spanner	Proposed real–time global leader board I would prefer Memorystore - Redis
Batch - Cloud Storage > Dataflow > BigQuery **Streaming - Pub/Sub > Dataflow > BigQuery**	Enable advanced analytics capabilities Explore Looker for advanced insights
Cloud Build, Spinnaker and/or Jenkins	CI/CD

Case Study - Mountkirk Games - Discussion - 2

In28 Minutes

Service	Discussion
Cloud Spanner + Memorystore	Match history database with regional in-memory caching
Cloud Bigtable	Time series events
Cloud Firestore	Player database
Cloud BigQuery, AI Platform	Datawarehouse and Intelligence
References	***https://www.youtube.com/watch?v=dAKAWprvbso*** ***https://www.youtube.com/watch?v=OF8EtzQ4e6c*** ***https://cloud.google.com/architecture/hybrid-multi-cloud-game-infrastructure*** ***https://cloud.google.com/architecture/gaming-backend-databases*** ***https://cloud.google.com/architecture/gaming-analytics***

Case Study - TerramEarth - Overview

In 28 Minutes

- **Manufactures heavy equipment** for the mining and agricultural industries
  - **Mission**: Build products that make their customers more productive
  - 500 dealers and service centers in 100 countries (**Global audience**)
  - 2 million TerramEarth vehicles with 20% growth year on year
    - Sensors capture telemetry data
    - Subset of critical data transmitted in real time
      - Rest of data collected, compressed, and uploaded daily when the vehicles return to home base
      - 200 to 500 MB of data per vehicle per day
- **Current Architecture**
  - **Google Cloud**
    - Data aggregation and analysis infrastructure
    - Web frontend for dealers and customers (stock management and analytics)
  - **Private data centers**
    - Legacy inventory and logistics management systems
    - Two main manufacturing plants send sensor data to private data centers
    - Multiple network interconnects to Google Cloud

Case Study - TerramEarth - Requirements

In 28 Minutes

- **Requirements:**
  - **Current:**
    - Provide best-in-class online fleet management services to customers
      - Improve operations of dealerships
      - **Minimize vehicle downtimes** (Detect failures and rapidly ship parts to dealerships for just-in-time repair)
    - Autoscaling, DevOps and SRE Elements
      - Improve & standardize tools for app and network monitoring, troubleshooting
      - Modernize **CI/CD pipelines** for **container-based workloads**
      - Allow remote developers to be productive
    - **Flexible & scalable platform for developers to create custom APIs for dealers & partners**
      - Create a new abstraction layer for HTTP API access to legacy systems (Enable gradual move to cloud)
      - Self-service portal for internal & partner developers to create new projects, request resources for data analytics jobs, and centrally manage access to the API endpoints
    - Use **cloud-native solutions** for keys and secrets management
  - **5-year strategic plan:**
    - Create a partner ecosystem of new products by enabling access to our data
    - Increase autonomous operation capabilities of vehicles
    - Path to move legacy systems to the cloud

Case Study - TerramEarth - Discussion - 1

In28 Minutes

Service	Discussion
IOT Core > Pub Sub > Dataflow > BigQuery > (AI Platform, AutoML, BigQuery ML)	Minimize vehicle downtimes - just-in-time repair Predictive Maintenance => ***https://www.youtube.com/watch?v=0VGBaaDKwbM***
Cloud Storage > Dataflow > BigQuery	Batch Process
Apigee	Flexible & scalable platform for developers to create custom APIs Create a new abstraction layer for HTTP API access to legacy systems (Enable gradual move to cloud) Self-service portal for internal & partner developers
Cloud Build, Spinnaker, Jenkins	Modernize **CI/CD pipelines** for **container-based workloads**
Google Workspace, Virtual desktops through partners	Allow remote developers to be productive

Case Study - TerramEarth - Discussion - 2

Service	Discussion
Secret Manager	Use **cloud-native solutions** for keys and secrets management
VPC Flowlogs, Cloud Monitoring, Cloud Logging etc..	Improve & standardize tools for app and network monitoring, troubleshooting
Kubernetes - GKE	Container-based workloads - scalable
Dedicated Interconnect	Networking(high data volume)