PROFESSIONAL CLOUD ARCHITECT

EHR Healthcare

Company overview

Healthcare: security, regulations, compliance

Global?

EHR Healthcare is a leading provider of electronic health record software to the medical industry. EHR Healthcare provides their software as a service to multi-national medical offices, hospitals, and insurance providers.

Containers, microservices
Artifact/Container Registry,
GKE, Cloud Run

Solution concept

Moving to DevOps, Agile, CI/CD? Source repositories, Cloud build

Due to rapid changes in the healthcare and insurance industry, EHR Healthcare's business has been growing exponentially year over year. They need to be able to scale their environment, adapt their disaster recovery plan, and roll out new continuous deployment capabilities to update their software at a fast pace. Google Cloud has been chosen to replace their current colocation facilities.

CI/CD

Moving to GCP, so might not be there yet. They need GCP for the agility and scalability it offers.

Existing technical environment

EHR's software is currently hosted in multiple colocation facilities. The lease on one of the data centers is about to expire.

Ah, so pressure to move. Lift-and-shift? Move-and-improve? Rearchitect? All of the above?

Customer-facing applications are web-based, and many have recently been containerized to run on a group of Kubernetes clusters. Data is stored in a mixture of relational and NoSQL databases (MySQL, MS SQL Server, Redis, and MongoDB).

GKE or possibly Cloud Run

Cloud SQL
Spanner?

Memory Store

Firestore, though not exact fit like others.

EHR is hosting several legacy file- and API-based integrations with insurance providers on-premises. These systems are scheduled to be replaced over the next several years. There is no plan to upgrade or move these systems at the current time.

Their Identity Provider (IdP)

Users are managed via Microsoft Active Directory. Monitoring is currently being done via various open source tools. Alerts are sent via email and are often ignored.

Cloud Monitoring... better alerting, perhaps other/better notification channels?

Business requirements

- On-board new insurance providers as quickly as possible.
- Provide a minimum 99.9% availability for all customer-facing systems.
  HA, 3-nines Remember, they are web apps
- Provide centralized visibility and proactive action on system performance and usage.
  Cloud Monitoring: Better monitoring, alerting, and dashboards
- Increase ability to provide insights into healthcare trends.
  Data analytics. Huge area. BigQuery, Looker, ...
- Reduce latency to all customers.
  Multi-regional setup, globally load balanced
- Maintain regulatory compliance.
  Duh, healthcare...
- Decrease infrastructure administration costs.
- Make predictions and generate reports on industry trends based on provider data.
  Looker for BI? Predictions, like ML?

Provider data, where is it? How do we get it? Analyze it?

Technical requirements

- Maintain legacy interfaces to insurance providers with connectivity to both on-premises systems and cloud providers. VPN? Interconnect? The on-prem isn't moving...
- Provide a consistent way to manage customer-facing applications that are container-based. More Cloud Run, GKE, consistancy
- Provide a secure and high-performance connection between on-premises systems and Google Cloud.

For sure, VPN and/or Interconnect, the high performance would lean more towards interconnect, depending on details.

- Provide consistent logging, log retention, monitoring, and alerting capabilities.
  Cloud Logging, Cloud Monitoring (with alerting)
- Maintain and manage multiple container-based environments.
  Hmmm, Anthos? GKE/Cloud Run? GKE Enterprise?
- Dynamically scale and provision new environments.
  GKE and Cloud Run both scale well, provision new? Terraform perhaps?
- Create interfaces to ingest and process data from new providers.
  Cloud Endpoints (legacy), API Gateway?, Apigee?
  or more Pub/Sub, Dataflow, like that? or both?

Executive statement

Big boss... always worth a look

Our on-premises strategy has worked for years but has required a major investment of time and money in training our team on distinctly different systems, managing similar but separate environments, and responding to outages. Many of these outages have been a result of misconfigured systems, inadequate capacity to manage spikes in traffic, and inconsistent monitoring practices. We want to use Google Cloud to leverage a scalable, resilient platform that can span multiple environments seamlessly and provide a consistent and stable user experience that positions us for future growth.

General, but always good to know the problems they are having, and why they are moving to GCP!