Infrastructure design for availability and resilience
Introduction

Organizations across industries such as retail, the public sector, media, finance and health care depend on Google Cloud to deliver mission-critical services. Whether it’s the launch of a global-scale game, processing retail transactions during Black Friday and Cyber Monday, serving real-time payments and trading transactions, or delivering key public services to citizens in times of crisis, customers around the world trust Google Cloud to be resilient, available, performant, and secure.

Google Cloud’s customers need insight into how the services they depend on are designed and operated to be resilient, available, and able to protect them from disruption. To deliver a reliable platform, Google Cloud leverages the technology, experiences, Site Reliability Engineering (SRE) culture, and expertise that Google has built over 20 years of operating global internet services at scale. The platform lets customers take advantage of the same infrastructure that supports Google's planet-scale products and services, including the following products with 1B+ active daily users: Android, Chrome, Gmail, Google Drive, Google Maps, Google Search, the Google Play store, and YouTube.

In this paper, we focus on how Google Cloud builds resilience and availability into its core infrastructure and services, from design through operations. We also explore the shared responsibility between Google and our customers—how customers can build on top of the core services we provide to gain the level of availability and resilience they need to run their businesses and meet their regulatory and compliance obligations.

For more on Google Cloud’s performance and security, see the following:
- Measure and compare performance
- Building Secure & Reliable Systems
- Google Infrastructure Security Design Overview

This paper is organized into five sections that showcase Google’s commitment to continuous availability:
1. Introduction
2. Google Cloud Platform Core Services
3. Key Customer Facing Services
4. Operations, Stress-Testing, and Validation
5. Resilience and Availability as a Shared Responsibility

Disclaimer

This whitepaper applies to Google Cloud products described at cloud.google.com. The content contained herein represents the status quo as of the time it was written. Google’s security policies and systems may change going forward, as we continually improve protection for our customers.
Google Cloud Platform Core Services

Infrastructure resilience and availability: leveraging Google technology, innovations and operational expertise.

Three of Google’s key principles in design, architecture, and operations are:

- **Fast is better than slow.**
- **Proactively identify dependencies and eliminate single points of failures and vulnerability before they matter.**
- **Automate to eliminate toil, manual effort, and errors.**

**Fast is better than slow**

Google systems are designed for minimum latency and maximum performance wherever possible. Google designs, builds, and deploys proprietary, purpose-built hardware and software across its infrastructure to achieve these performance objectives.

**Proactively identify dependencies and eliminate single points of failure and vulnerability before they matter**

Google’s core services are globally distributed. Google engineers proactively search for dependencies in their systems and redesign to eliminate them. Finally, Google prides itself on a blameless post-mortem culture. Engineers learn from each and every failure, continuously updating their understanding of service dependencies and the root cause of errors.

**Automate to eliminate toil, manual effort, and errors**

Changes to Google Cloud configurations, infrastructure, and services are made programmatically through an audited software development lifecycle and management process. Google Cloud executes and operates its planet-scale services as containerized, stateless microservices orchestrated by Borg, Google’s cluster manager. (Key parts of Borg are open-sourced as Kubernetes.) By building Google services on a foundation of immutable microservices, the computational fleet is consistent, scalable, resilient, and highly available.

Combining these principles with Google Cloud infrastructure—which is based on wholly-owned, purpose-built hardware and software systems—Google is able to exert high levels of control over end-to-end performance, resilience, availability, security, and operational behavior.
Examples of key core systems, services, and patterns

Many Google services, Google Cloud and otherwise, leverage the same Google innovations and proven and robust architecture patterns for their serving path. These innovations include Google's Identity system, the Google Cloud Network, the Google Front End (GFE), Borg orchestration and control, Spanner, and common self-service UI and API surfaces.

Google Identity system

Google's infrastructure provides a rich identity management system for internal identities, including workflows for approval chains, logging, and notification. For example, identities can be assigned to access control groups with two-party control; when one engineer proposes a change to a group, another engineer (an administrator of the group) must approve the change.

Google's Identity system also allows secure access management processes to scale to millions of services running on Google's infrastructure. The system is globally distributed and serves users, services, applications, and devices. Google has analyzed its dependencies and employs redundancy, compensating controls, and back-up systems to limit the impact of disruptions for customers and to maintain resilience and availability.

Systems across Google, including Google Cloud, Google Search, Android, Maps, YouTube, Google Drive, Chrome, Gmail, and Android PlayStore, leverage this global-scale system for user, service, application and device identity.

The Google Cloud Network

Google's network provides high performance, scale, and redundancy for customers through globally distributed entry points. Google's cloud network architecture consists of 22 regions, 67 zones, and 140 network locations, and is available in 200+ countries across six continents. Google continues to invest in network capacity, performance, and reach by building out new regions, zones, and network locations.

Google's high-performance network employs encryption in transit and is globally flat to minimize transformations and interfaces in the communication path. This builds in redundancy and security, minimizes latency, and maximizes throughput performance and bandwidth for customers. Once Google Cloud customers connect to the Google network, infrastructure from the entry point to the server is owned, operated, and controlled by Google, enabling Google to provide consistent operational control, resilience, availability, security, and performance across its network.
The Google Front End (GFE)

The GFE provides a fully distributed but logically centralized termination point for all ingress traffic to Google Cloud. The GFE acts as the access point for all services and APIs. It supports anycast networking, which provides multipath redundancy and resilience, as well as built-in DDoS protection. Anycast allows customer traffic to gracefully roll over to the nearest instance of the GFE in the event of zonal or regional disruption.

Borg orchestration and control

All Google Cloud services are built as sets of stateless microservices orchestrated and controlled by Borg. Borg supports high-availability applications through a control plane with runtime features that minimizes fault-recovery time and applies scheduling policies to reduce the probability of correlated failures. The stateless microservices that compose Google Cloud services are autoscaling, globally distributed across data centers, and architected for resilience. They can be stopped and started without loss of data, so Google Cloud can maintain and update services without customer-visible downtime, disruption, or the need for scheduled maintenance windows.

Spanner

Spanner, Google’s unique, globally distributed but strongly consistent database service, provides foundational resilience and reliability for Google Cloud services. While other cloud providers deliver reliability through loosely consistent systems, Spanner enables core Google Cloud services to be designed and run with a simple operational model which has strong global consistency guarantees, without sacrificing resilience and availability.

Self-service UI and API surfaces

Google’s software infrastructure is built on API-centric stateless microservices. The serving paths for both UI and API requests travel from the user to the GFE, and then to the corresponding set of stateless microservices for response and action. As with other Google Cloud services, the Cloud Console that customers interact with is also built from a set of stateless microservices managed by Borg, which ensures a resilient and highly available control plane.

---

¹Throughout this paper, “Spanner” refers to Google’s internal database service and not the customer-facing version of the product Cloud Spanner.
Architecture overview for key Google Cloud services

The diagram below shows the high level components that comprise the serving path for key Google Cloud services. UI and API requests travel from the user over the public internet to the Google point of presence (POP) closest to the user, and terminate at the GFE. These requests then travel over Google's proprietary global network to the destination needed to serve the request. The serving resources are organized as redundant, auto-scaling, and stateless microservices managed by Borg. Compute resources leverage Spanner for database services, then send their response back through the Google network, GFE, and public internet back to the user.
Key customer facing services

Introduction

Google Cloud is responsible for making sure that our products and services are resilient and available, and for empowering customers to build on them to meet the availability, resilience and disaster recovery requirements of their businesses. Google Cloud architects and operates services to include separation, robust replication, and isolation against single-point failures. Throughout the design, build, deploy, and run process, Google proactively performs resilience testing, dependency identification, and mapping to find potential single points of failure, and then works proactively to correct any issues to minimize the impact of disruptions on customers. Services at Google are continuously monitored for their availability and graded against their SLO metrics. Google manages risk and reliability of services through the use of service-level error budgets. In this section, we provide additional insight into two of the key services that customers can leverage and build upon: Identity and Access Management (IAM) and Key Management Services (KMS).

Identity and Access Management (IAM)

What is IAM?

[Cloud Identity and Access Management](https://cloud.google.com/identity-and-access-management), or “IAM”, provides fine-grained access control and auditability for Google Cloud Platform (GCP) resources. IAM leverages the globally distributed, resilient, and available [Google Identity system](https://cloud.google.com/identity) discussed in the previous section, and is built on [Spanner](https://cloud.google.com/spanner), Google's globally distributed database.

With IAM, client services automatically enforce policies in the most logical region. Policies are then automatically replicated first to a global IAM instance and then to regional instances to ensure global consistency. This replication also provides increased availability, since it ensures that policies will be enforced even if an individual region experiences an outage. These capabilities exemplify the ease and reliability of using Google Cloud—customers can manage resources and enforce policies globally without global outage risk. Currently, over 300 Google Cloud services use IAM for authorization.

IAM’s availability

IAM must be consistently available for Google Cloud Platform services to be accessed—there is no good time to be down. There is zero planned downtime; IAM never shuts down for scheduled maintenance. IAM, similar to all Google Cloud core services, is built as a set of stateless microservices managed by Borg, which is designed such that we can start and stop microservices without loss of data, eliminating the need for scheduled maintenance windows or downtime.
Global consistency with regional availability

IAM allows users to maintain global consistency through a global IAM instance that maintains copies of all IAM policies, even ones set in a specific region, and replicates those policies to regional IAM instances. This combination of the global instance and regional replication ensures that customers don’t have to worry about whether their IAM policies are set in the correct region. In addition to ensuring global consistency, regional replication ensures high availability for IAM policy enforcement by ensuring that regional failures will not impact availability outside of their region. Individual regions also use zonal replication to ensure that the regions themselves are highly available.

The diagram below demonstrates IAM requests’ built-in redundancy and multiple paths for graceful degradation.

Diagram of runtime CheckPolicy() call highlighting usage of regional instances to ensure reliability:
Cloud Key Management System (KMS)

What is Cloud KMS?

Cloud KMS is a cloud-hosted key management service that lets customers manage cryptographic keys for cloud services in the same way they manage keys on-premises. With Cloud KMS, customers can generate, use, rotate, and destroy AES256, RSA 2048, RSA 3072, RSA 4096, EC P256, and EC P384 cryptographic keys. Cloud KMS is also integrated with Cloud Identity and Access Management and Cloud Audit Logs, so customers can manage permissions on keys and monitor their use.

Cloud KMS can be used in conjunction with services like BigQuery, Cloud Storage, Cloud Build, Dataproc, Pub/Sub, Compute Engine, Container Registry, Cloud SQL, and Google Kubernetes Engine to control and manage the encryption of data in each of these services. Google Cloud continues to expand the number of services that integrate with Cloud KMS with Customer Managed Keys (CMEK).

Cloud KMS availability

When using Cloud KMS with a data service like Google Cloud Storage or Google BigQuery, as with IAM, there is zero planned downtime; Cloud KMS never shuts down for scheduled maintenance. Because Cloud KMS, like all Google Cloud core services, is built as a set stateless microservices managed by Borg, which is designed such that we can start and stop microservices without loss of data, eliminating the need for scheduled maintenance windows or downtime. As a result, Cloud KMS provides 99.5% uptime to customers for Encrypt, Decrypt, and Sign operations.

Global consistency with regional availability

Cloud KMS has an architecture based on Borg cells, sets of machines that offer high redundancy to ensure continuous availability. Data that KMS depends on is replicated to each cell within the region. Within each cell, data is replicated across three separate tasks. Multiple cells within a region can fail, but the surviving cells continue serving KMS requests. Cloud KMS uses Spanner to store key material, and relies on core Google systems for identity and access management. Beyond that, a Cloud KMS task has no dependencies outside of its cell.

To deliver high authorization availability, KMS replicates the global state to specific regions. This means that regional failures should not impact availability outside of their region. Each KMS region leverages regional data stores that are replicated across multiple zones in the region, meaning that zonal failures should also not impact availability.
Resilience and availability examples for related services

Cloud KMS works with other Google Cloud services, like Google Cloud Storage and BigQuery, through an integration that is architected for resilience and continuous availability. This integration, called customer-managed encryption keys (CMEK), lets customers encrypt data with keys that they manage in Cloud KMS, giving finer control over encryption at the storage level.

**Google Cloud Storage (GCS)**

To provide regional isolation, CMEK integration with Google Cloud Storage only allows customers to use encryption keys created in the same region, dual-region, or multi-region as their GCS bucket. For example, if a bucket is located in us-east1, any Cloud KMS key encrypting objects in that bucket must also be created in us-east1.

Zonal failures should not impact Cloud KMS availability, so CMEK integration is always available to all GCS zones in a region. Customers can help ensure CMEK availability even in the event of a regional failure by following GCS location guidance and Google Cloud's guidelines for geographic management of data.

In the event of a full datacenter failure in a dual- or multi-region, Cloud KMS still serves key requests to GCS from the other datacenters in the region.

Learn more about CMEK in GCS.

**BigQuery**

Just like GCS, CMEK integration with BigQuery only lets customers use encryption keys created in the same region, dual-region, or multi-region as their BigQuery dataset. For example, if a dataset is located in us-east1, any Cloud KMS key that encrypts that dataset must also be created in us-east1.

Customers can help ensure CMEK availability even in the event of a regional failure by following BigQuery location guidance and Google Cloud's guidelines for geographic management of data.

Learn more about CMEK in BigQuery.
Operations, stress testing, and validation

Introduction

In the previous sections, we covered how Google Cloud designs, architects, and builds core services for resilience and availability. In this section, we will cover two examples of how Google Cloud complements its robust design and execution with ongoing operations built to maximize reliability, resilience, and availability.

Site Reliability Engineers and SRE culture

Site Reliability Engineers (SREs) apply engineering practices to the management of large-scale systems. They work with product development teams to jointly agree on Service Level Objectives. They develop practices and tools that help design, build, and run large-scale systems scalably, reliably, and efficiently. SREs seek to balance reliability versus feature velocity and seek to maintain production reliability and stability while removing obstacles to shipping new features.

Error budgets are one of the mechanisms that provide for taking risks and accelerating feature velocity. Per the SRE Book, when services are operating within their error budgets, the corresponding product development teams can take more risks, deploy new features, and accelerate their launch velocity. As a service consumes its error budget, product development teams must naturally slow down their launch velocity and execute more testing to reduce the risk of using up their remaining budget. If a service exhausts its error budget, then SRE teams have the authority to stop launches, causing product teams to stop feature work and focus solely on reliability improvements until they are back under their error budget. The use of error budgets enables product development teams to be self-policing and ensures that product teams and SRE teams jointly share the responsibility for resilience and availability.

Not only do SREs unite software and systems engineering skill sets, they focus on the whole lifecycle of services—from inception and design, through deployment, operation, refinement, and eventual peaceful decommissioning. SRE teams prize automation and hate “toil”. They automate routine, error-prone manual tasks and collaborate with product development teams to identify and eliminate key dependencies and single points of failure. To further improve resilience, SRE teams work with product teams to define Disaster Recovery Testing (DiRT) to proactively find vulnerabilities, single points of failure, and test resilience.

In the event of an outage, developers and SREs alike follow a strict Incident Response and Management protocol and process, and respond afterwards by collaborating on blameless post-mortems that include action items designed to prevent future occurrences. SREs play a critical role in enabling Google Cloud to deliver and continuously operate resilient and available services and to minimize the impact and disruption from outages.
Disaster Recovery Testing - DiRT

Google runs annual, company-wide, multi-day Disaster Recovery Testing events (DiRT) to ensure that Google's services and internal business operations continue to run during a disaster. DiRT was developed to find vulnerabilities in critical systems by intentionally causing failures, and to fix those vulnerabilities before failures happen in an uncontrolled manner. DiRT tests Google's technical robustness by breaking live systems and tests our operational resilience by explicitly preventing critical personnel, area experts, and leaders from participating. All generally available services are required to have ongoing, active DiRT testing and validation of their resilience and availability.

DiRT exercise examples

To prepare for a DiRT exercise, Google employs a consistent set of rules around prioritization, communication protocols, impact expectations, and test design requirements, including pre-reviewed and approved rollback plans. DiRT exercises and scenarios not only force technical failures in the service itself, but also can include designed failures in process, availability of key personnel, supporting systems, communications, and physical access. DiRT validates that the processes in place actually work in practice. It also ensures that teams are pre-trained and have experience they can draw upon during actual outages, disruptions, and man-made or natural disasters. Here are a few examples of past DiRT testing scenarios from IAM and Cloud KMS:

IAM
IAM SREs have run DiRT exercises that test a variety of scenarios, including the effect of a degraded IAM global instance on IAM regional instances. They also tested the robustness of IAM synchronization jobs that assure regional instances receive updates made to the global instance.

Cloud KMS
Cloud KMS SREs have run DiRT exercises that test a variety of scenarios, including database loss, task failure, and dependency loss. In one recent test, the team caused a regional outage for a logging service that KMS depends on. The outage didn't impact KMS availability and no logs were lost, but the test pointed out ways the team could further improve procedures and streamline their alerts.
Resilience and availability as a shared responsibility

Introduction

Ultimately, the resilience and availability of applications and workloads built on top of Google Cloud Platform services is a shared responsibility. In the previous sections, we covered how Google Cloud Platform builds and operates the services customers depend on. In this final section, we cover some of the key considerations for building resilience and availability for workloads, and how we support customers throughout the process. As covered in the previous sections, we believe that in order to build resilient and available systems, customers need more than just great design and architecture— to truly operate confidently at a global scale, customers need to consider planning, process, training, and people as well.

Key considerations for high availability and disaster recovery

When designing systems and services for resilience and availability, important considerations include:

- Clearly define resilience and availability objectives and metrics
- Understand the dependencies and single points of failure in the planned services and workloads
- Account for and plan for differences in geographies
- Assess end-user tolerance for service degradation
- Accommodate regulatory and policy compliance constraints
- Consider whether to use a single cloud, multi-cloud, and/or hybrid environment
- Put the right operational processes in place
- Prepare and proactively train staff on the processes and procedures through “live-fire” exercises
- Review and leverage high availability features and building blocks that are built in to Google Cloud Platform
- Determine how to enable continuous learning and incorporate feedback to improve resilience and availability on an ongoing basis
Learn more

Google Cloud has technical experts that help customers architect and build highly available systems using Google Cloud products and components by recommending architectures and configurations. See the following resources to get started.

Disaster Recovery Planning Guide

Overview of the high availability configuration for Cloud SQL

Process, procedures and culture guidance - SRE Book

How Google Cloud Supports Customers

Google Cloud directly supports customers by providing independent compliance validation for Google Cloud services, visibility into the status and health of Google Cloud services, and customer support channels that provide expert guidance.

Third-party validations and certifications

Google Cloud maintains an ongoing certification process for core platform services to obtain independent third-party validation. Current compliance statuses are updated in the Compliance resource center.

Service Availability Dashboards

Google Cloud provides live visibility into service status and outages through the Google Cloud Status Dashboard. In addition, customers can review which products and services are available by location on the Cloud locations page.

Customer Support

Google Cloud has a range of experts to help customers throughout the design, build, deploy and operate lifecycle. Solutions and best practices guides are available on the Google Cloud Solutions page. Customers can engage with Google experts on the Consulting Services page for assistance in the build and deploy phases. Google also provides support services and a support hub for customers to leverage during operations.
References

2017 “Calculus of Service Availability” Ben Traynor, Mike Dahlin et al

2016 SRE Book - core process and culture around availability and reliability

2020 SRS Book - Building Secure & Reliable Systems

2016: SRE Book: Chapter 19: Load Balancing at the Front End

2013 “Spanner: Google’s Globally Distributed Database James Corbett, Jeff Dean et al.

2012 “Weathering the unexpected - Kripa Krishnan