Google Cloud Architecture Framework

Google Cloud Architecture Framework

In 28 Minutes

- **Best practices** and **implementation recommendations** to help you design your Google Cloud deployment
- **First Step**: Focus on designing robust, secure, and scalable systems
- **4 Principles**:
  - Operational excellence
  - Security, privacy, and compliance
  - Reliability
  - Performance and cost optimization

Google Cloud

Principle 1: Operational Excellence

In 28 Minutes

- "Efficiently running, managing & monitoring systems that deliver business value"
- Strategies:
  - Automate build, test, and deploy
  - Monitor business objectives metrics
  - Conduct disaster recovery testing

Google Cloud

Operational Excellence - Best practices

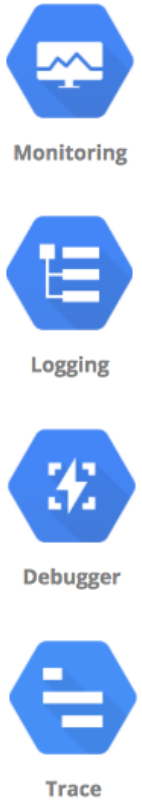
In 28 Minutes

- Increase software development and release velocity
  - **Release Engineering**
    - Frequent small releases
  - **Automation**
    - Static code analysis and security scans
    - Automate Testing (Unit, Integration, System, Load, Security Testing etc..)
    - Automate build and release pipeline
      - A/B or canary testing
  - **Services:**
    - **Cloud Source Repositories**: Fully-featured, private Git repository
      - Similar to Github
    - **Container Registry**: Store your Docker images
    - **Cloud Build**: Build deployable artifacts (jars or docker images) from your source code and configuration

Operational Excellence - Best practices - 2

In 28 Minutes

- Monitor system health and business health
  - **Understand four golden signals**
    - Latency - How fast are you responding to your users?
    - Traffic - How does the load on your system look?
    - Errors - Are any requests failing?
    - Saturation - What's the utilization levels of your resources?
  - **Logging**: Use Cloud Logging and export to Cloud Storage, BigQuery, and Pub/Sub depending on your needs
  - **Metrics**: Define and capture metrics SLIs, SLOs etc
  - **Monitoring**: Cloud Monitoring can aggregate metrics, logs, and events
    - Define alerts and custom metrics to identify problems and resolve them quickly
    - You can create visual dashboards as well
    - Generate actionable alerts
  - **Key Services**: Cloud Monitoring, Cloud Logging, Cloud Debugger, Error Reporting, Cloud Trace and Cloud Profiler



Operational Excellence - Best practices - 3

In 28 Minutes

- **Design for disaster recovery**
  - Create **well-defined disaster recovery (DR) plan**
    - Define **Recovery time objective (RTO)** and **recovery point objective (RPO)**
    - Consider everything in your DR plan: Applications (compute etc), Data(databases etc), Network infrastructure, Bandwidth, Facilities
    - Use features provided by Google Cloud:
      - Use Global network to build Redundancy
      - Managed services make Scalability easy
  - **Regularly test your DR plan**
  - **Example features** you can make use of:
    - Schedule Persistent Disk snapshots for your VMs and copy them across regions
    - Enable Live Migration to keep your VMs running even when there is software or hardware maintenance
    - Use Cloud DNS to switch from primary to backup

Google Cloud

Principle 2: Security, Privacy and Compliance

In 28 Minutes

- Plan your security controls, privacy, and meet your compliance needs

Google Cloud

- **Strategies**
  - Implement least privilege with identity and authorization controls
  - Build a layered security approach
  - Automate deployment of sensitive tasks
  - Implement security monitoring

Security, Privacy and Compliance - Best practices

In 28 Minutes

- **Manage Authentication and Authorization**
  - Follow **Identity and Access Management (IAM) Best Practices**
    - Grant appropriate roles
    - Understand when to use service accounts
      - Treat each app component as a separate trust boundary
        - Create separate service account for each component/service with the minimum permissions needed
    - Use Organization Policy Service (what is allowed in your organization?)
    - Use Cloud Asset Inventory (Track your inventory)
    - Use Cloud Audit Logs to audit IAM policy changes and service accounts
- **Implement Compute Security Controls**
  - Use Private IPs as much as possible
  - Create hardened VM images (OS + minimum software)
  - Use Shielded VMs to prevent remote attacks, privilege escalation, and malicious insiders
  - Enable node auto-upgrades for GKE clusters

Google Cloud

Security, Privacy and Compliance - Best practices - 2

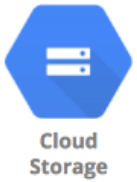
In 28 Minutes

- **Secure the network**
  - Use a carefully designed custom VPC (DO NOT use the default VPC)
  - Isolates workloads into individual projects by creating one VPC per project
  - **Control ingress and egress** network traffic using **firewall rules**
  - Run advanced security and traffic inspection tools
  - Use **Security Command Center** to analyze the security of your infrastructure
  - Use **Network Intelligence Center** to evaluate your network topology and architecture

Security, Privacy and Compliance - Best practices - 3

In 28 Minutes

- Implement **data security controls**
  - (Default) Google Cloud encrypts customer data stored at rest
    - Use customer managed or customer supplied keys based on your needs
  - Use **Object Versioning** in Cloud Storage for sensitive data
    - Use Object Lifecycle Management to reduce costs
    - Achieve compliance with retention policies using Bucket Lock
    - Use Signed URLs to give time-limited read or write access to an object
  - **Implement data security**
    - Use Cloud Data Loss Prevention (DLP) API to classify, mask, tokenize, and transform sensitive data (ensure data privacy for sensitive data elements like credit card numbers, names, SSNs, phone numbers etc..)
- **Audit your infrastructure with audit logs**
  - Use Cloud Audit Logs
    - Export to Cloud Storage, BigQuery, or Pub/Sub based on your needs
  - Enable Access Transparency logs (trace Google support team actions)



Principle 3: Reliability

In 28 Minutes

- **Most important feature of any application/service:**
  - **Keys:**
    - Measurable reliability goals
    - Architect for scalability, high availability, and automated change management
    - Self-healing and observability are important
    - Quick recovery from failures using automation as much as possible
  - **Strategies:**
    - Define KPIs, SLOs and SLIs
    - Create redundancy (Prefer horizontal scalability)
    - Make incremental changes
      - Include rollback capability
    - Instrument systems for observability
    - Automate detection of failure
    - Document and automate emergency responses
    - Test failure recovery
    - Reduce toil (Eliminate manual work)

Google Cloud

Reliability - Best practices

In 28 Minutes

Google Cloud

- **Define reliability goals**: Service Level Objectives & Error Budgets
  - Clearly define SLIs, SLOs, SLAs and Error budgets
- **Build observability** into your infrastructure and applications
  - Monitoring, logging, tracing, profiling, debugging etc.
- **Design for scale and high availability**
  - Design a multi-region architecture with failover
  - Eliminate scalability bottlenecks
    - Prefer horizontal scalability
  - Degrade service levels gracefully
    - Static web pages when dynamic web site is down
    - Temporarily disabling data updates
  - Implement exponential backoff with jitter
    - Exponentially increasing wait time before retry
  - Predict peak traffic events and plan for them

Reliability - Best practices - 2

In28 Minutes

- **Build flexible and automated deployment capabilities**
  - Ensure that every change can be rolled back
  - Spread out traffic for timed promotions and launches
  - Implement progressive rollouts with canary testing
  - Automate build, test, and deploy
- **Build efficient alerting**
- **Build a collaborative process for incident management**
  - Reduce Mean Time to Detect (MTTD): Alert teams at the right time
  - Reduce Mean Time to Mitigate (MTTM) and Mean Time to Recovery (MTTR): Properly documented and well-exercised incident management plan
  - Increase Mean Time Between Failures (MTBF): Build reliable systems
  - Blameless postmortem culture

Google Cloud

Principle 4: Performance & Cost Optimization

In 28 Minutes

- **Strategies**
  - Evaluate performance requirements
  - Use scalable design patterns
  - Identify and implement cost-saving approaches

Google Cloud

Performance and Cost Optimization - Best practices

In 28 Minutes

- **Use autoscaling and data processing**
  - Autoscale Compute Engine VMs with Managed instance groups (MIGs) - Uses an instance template
  - Enable Cluster autoscaler and Pod autoscaling (Horizontal Pod Autoscaler) on Google Kubernetes Engine Clusters
  - Try Serverless options: Cloud Run, App Engine, Cloud Functions, Dataproc and Dataflow
  - Use Google Cloud Load Balancers to provide a global endpoint.
- **Use GPUs and TPUs to increase performance**
  - Use GPUs to accelerate machine learning and data processing workloads
  - Use TPUs for massive matrix operations performed in your machine learning workloads

Performance and Cost Optimization - Best practices - 2

In 28 Minutes

- **Identify apps to tune**
  - Use Cloud Trace, Cloud Debugger, and Cloud Profiler to gain insights into your apps
  - Instrument apps to identify inter-service communication latencies or identify bottlenecks
- **Analyze your costs and optimize**
  - Export Billing to BigQuery to analyze your billing data
    - Use Google Data Studio to visualize
  - Understand Sustained use discounts
  - Make use of Committed use discounts for predictable long term workloads
  - Use Preemptible VMs for non critical fault tolerant workloads
  - Use Cloud Storage Object Lifecycle Management to reduce storage costs