Architects & Architecture Must Know

Architect : Understand Business Requirements

In 28 Minutes

- **Business Requirements** : Define what your business needs
- Examples of Business Requirements:
  - **Reduce costs** (Understand Capital Expenditure vs Operational Expenditure)
    - Managed Services
    - Autoscaling
    - Premptible VMs
  - **Increase pace of innovation**
    - Evaluate and adopt emerging processes like DevOps (CI/CD) and SRE
    - Evaluate and adopt emerging architectures like Microservices
  - Reduce mean time to recovery
  - Improve compliance with regulations
  - More visibility into applications and infrastructure (metrics)
  - More intelligence from the available data

Google Cloud

Total Cost of Ownership(TCO)

In28 Minutes

- **Total Cost of Ownership(TCO)** includes:
  - Licensing Costs (Software + Hardware)
  - Computing Costs
  - Storage Costs
  - Networking Costs (Connection cost + Data Ingress + Data Egress)
  - Personnel Costs (Dev + Test + Ops + Business + ..)
  - Other Costs
    - Penalties for missed SLAs or Compliance needs
    - Third Party APIs
    - ....
- **When designing solutions**, ensure that you take Total Cost of Ownership(TCO) into account!

Google Cloud

Architect : KPIs for Business Requirements

In28 Minutes

- How can you measure **if you are meeting Business Objectives**?
- **Define KPIs (Key Performance Indicator)** where possible:
  - How is the business doing with respect to an objective?
  - Project KPIs
    - Example: Number of new customers in a week
    - Example: Percentage of virtual machines running in the cloud
  - Operational KPIs
    - Example: Operational cost per customer

Google Cloud

Defining Technical Requirements

In 28 Minutes

- Technical aspects that **your system must adhere to**
  - **Functional:**
    - Must use containers
    - Must use hardened Linux OS
    - Needs container orchestration
    - Must be auto scaling
    - Must be NoSQL database with flexible schema
    - Able to store huge volumes of archives at very low cost
    - Must use a private network (traffic should not go over internet)
  - **Non Functional**:
    - Availability
    - Scalability
    - Durability
    - Security

Google Cloud

Planning for High Availability

In28 Minutes

Service/Feature	Features / Best Practices
Geographical Distribution	Availability: Global > Multi-Regional > Regional > Zonal
Compute Engine	Live Migration Managed Instance Groups with AutoScaling and Health Checks (Auto healing) Distribute instances across regions/zones and distribute using Global Load Balancing
GKE	Multi master, Regional clusters with pod and cluster autoscaling
Managed Services	Use Managed Services like App Engine, Cloud Functions, Cloud Storage, Cloud Filestore, Cloud Datastore, BigQuery
Persistent Disks	Live resizing improves availability Use Regional Persistent Disks
Cloud Bigtable	Place clusters in different zones or regions (Each cluster belongs to a zone and you can create multiple clusters for high availability in the same instance)

Planning for High Availability - 2

In28 Minutes

Service/Feature	Features / Best Practices
Cloud Datastore	Use Multi-region locations
Cloud SQL	Use **HA configuration** (regional) - Cluster with primary instance and a standby instance is created Read replicas will NOT be promoted (So Read Replicas do NOT increase availability for Cloud SQL)
Network Tier	**Prefer Premium Network Tier** to Standard Network Tier
Hybrid Network Connectivity	Speed and Availability: Dedicated interconnect > Partner interconnect > VPN Have a backup connection. Example: VPN can be a backup for Dedicated interconnect or have multiple Dedicated interconnect connections

Planning for Scalability

In 28 Minutes

- **Highly Scalable Compute Engine Architecture** : VMs in MIG configured by instance template (and load balanced with Load Balancing)
  - (Remember) Unmanaged Instance Groups do not support Auto Scaling
- Use **Pod and Cluster Autoscaling** for GKE
- Be cautious with resources that cannot scale fast
  - Cloud SQL does NOT scale horizontally (only vertically)
- Stateful applications are more difficult to scale than Stateless applications
  - You can move state to a cache (like Memorystore) or a database
- **Local SSD** is least scalable (Max 8 per VM)
  - Data survives reboots and live migrations
  - But data is lost when VM is stopped or terminated!

Google Cloud

Planning for Scalability - 2

In28 Minutes

- **Persistent Disks:** Can be scaled horizontally and vertically
  - You can increase the size of PD while it is attached to a VM
  - You can attach new PDs (up to a max of 128) while a VM is running
- **Compute**:
  - Cloud Storage, App Engine and Cloud Functions are serverless (Auto Scaling)
- **Databases:**
  - Pub/Sub, BigQuery and Cloud Datastore are serverless (Auto Scaling)
  - BigTable, Cloud Spanner, Cloud SQL, Dataproc are **NOT serverless**
    - You need to choose the number of nodes in the cluster (and also the type of node for Cloud SQL and Dataproc)

Google Cloud

Planning for Security

In 28 Minutes

- **Confidentiality** - Only right people have the right access
  - Follow **IAM Best Practices**
  - Encryption at rest and in transit (along with physical controls)
    - (DEFAULT) GCP encrypts all data at rest
- **Integrity** - Protect data from unauthorized change
  - Follow **IAM Best Practices**
    - Role Based Access
    - Separation of duties
  - Hash verifications and digital signatures
- **Availability** - Systems/Data is/are available when users need them
  - Solutions: Firewalls, Redundancy, Automatic Failover etc
  - Protect from Denial of Service (DoS) attacks

Google Cloud

Implementing DDoS Protection and Mitigation

In28 Minutes

- **(DDoS) attack**: Attempting to bring your apps down with large scale attacks
- **Shared responsibility** between GCP and Customer
  - GCP provides certain features to protect your from DDoS attacks
    - **Anti-spoofing** protection for the private network
    - **Isolation** between virtual networks
    - App Engine (sits behind Google Front End) automatically protects you from Layer 4 and below attacks
      - Examples: SYN floods, IP fragment floods, port exhaustion, etc.
  - What you can do to protect apps from DDoS attacks?
    - **Reduce the attack surface**: Make use of subnetworks and networks, firewall rules and IAM
    - **Isolate** your internal traffic
      - Use Private IP Address (unless you need Public IP Addresses)
      - Use private load balancing for internal traffic
    - **Use Proxy-based Load Balancing**: Automatically protects you from Layer 4 and below attacks
    - Integrate Load Balancing with **Cloud Armor**
      - IP-based and geo-based access control
      - Enforce Layer 7 security policies on hybrid and multi-cloud deployments
      - Pre-configured WAF rules (OWASP Top 10)

Digital Signatures - Cloud KMS

In 28 Minutes

- When do we use Digital Signatures?
  - **Purpose 1**: Verification of the integrity of the signed data
  - **Purpose 2**: Non-repudiation if the signer claims the signature is not authentic
    - Sender cannot deny sending the message later

- Workflow:
  - Sender performs private key operation over the data to create a digital signature
  - Recipient uses the public key to verify the digital signature
    - If verification is unsuccessful, then the data has been altered
- Cloud KMS can be used to create an asymmetric key pair (public and private key pair) that supports digital signing
  - Provides APIs/commands to create digital signatures
    - `gcloud kms asymmetric-sign`
- Usecases: Validating code builds

Cloud Armor

In28 Minutes

- **Cloud Armor:** Protect your apps from denial of service and OWASP Top 10 attacks
  - Protects you from **common web attacks** (OWASP Top 10) like XSS (cross-site scripting) and SQL injection
  - Protect applications deployed in Google Cloud, in a hybrid deployment, or in a multi-cloud architecture
    - Integrates very well with Google Cloud Load Balancing
  - Provides **preconfigured security policies** (OWASP Top 10 risks etc)
    - Customize rules as per your needs
  - **Usecases**
    - Enable access for users at specific IP addresses with allowlists
    - Block access for users at specific IP addresses with denylists
    - Protect applications against OWASP Top 10 risks

Secret Manager

In28 Minutes

```
SecretVersionName secretVersionName = SecretVersionName.of(projectId, secretId, versionId)
AccessSecretVersionResponse response = client.accessSecretVersion(secretVersionName);
String secretValue = response.getPayload().getData().toStringUtf8();
```

- How do you **manage your database passwords, your API keys** securely?
- **Secret Manager** - Store API keys, passwords etc
  - Multiple versions of secrets
  - Automate rotation with Cloud Functions
  - Auditing with Cloud Audit Logs
  - Encrypted by default
- Best Practice: **Do NOT store credentials/passwords in code**
  - Use Secret Manager and access secrets in your application

Architect Responsibilities: Stakeholder Management

In 28 Minutes

- **Stakeholder:** "a person with an interest or concern in something"
  - Different stakeholders have different priorities/interests/degrees of influence:
    - CEO/CTO
    - Business Process Owner
    - Product Owner
    - Architecture Team
    - Scrum Master
    - Development Team
    - Testing Team
    - Security Team
    - Compliance Team

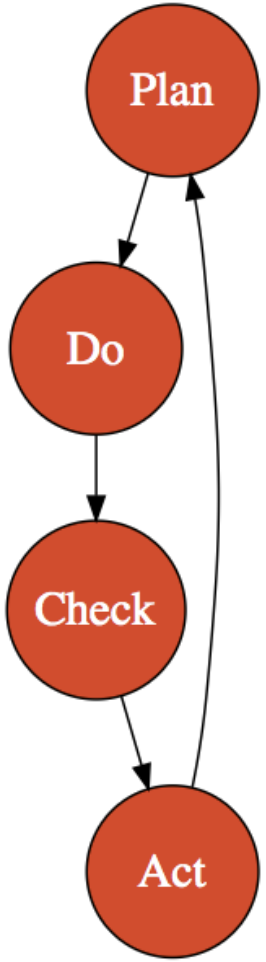
Google Cloud

- As an Architect, you need to **understand the needs of different stakeholders and design solutions** that meet them
  - Key to Stakeholder Management: **Early clear communication**
    - Identify stakeholders, their interests and build a plan to effectively communicate with them

Architect Responsibilities: Change Management

In 28 Minutes

- **Changes are inevitable**
  - People, Process and Technology (systems, hardware, and software)
  - But humans hate change
- **Change Management**: How do you deal with changes and reduce impact?
  - Understand the change:
    - Reason: Why make the change?
    - Risks: What are the risks?
    - Resources: Who is impacted? Who can help make the change? What is needed?
    - Responsibility: Who can make it happen?
  - Follow the cycle: Plan - Do - Check - Act



Architect Responsibilities: Business Continuity Planning

In 28 Minutes

- BCP: How to keep business running in face of disasters?
- **Disaster Recovery :** Parts of BCP focused IT operations
  - **Example**: Use cloud/on-prem as DR environment
  - **Example**: Have a backup network connection between cloud and on-prem
  - **Disaster Recovery Plan:** How should an enterprise respond to different types of disasters?
    - Includes People, Process and Technology (systems, hardware, and software)
    - Identifies critical services, personnel and plans
    - Define Recovery Time Objective (RTO) and Recovery Point Objective (RPO)
      - Design and Architect your solutions to meet your RTO and RPO objectives
  - DR plans should be **continuously tested** (Game days)
    - Are you able to restore a database from archive?
- **BCP = Disaster Recovery Plan** (Restore critical IT applications and systems) + **Business recovery** (Restore critical business processes)
  - Example: Outsourcing
  - Example: Temporary Office Space

Architect Responsibilities: Incident Management

In 28 Minutes

Google Cloud

- **Incident :** "unplanned event that causes a service disruption"
- **Goal:** "How to avoid incidents?" and "How to react to incidents quickly?"
- **How to avoid incidents?**
  - Monitoring and Alerting
    - Examples: Alerts : free disk space in Storage/Databases, CPU Utilization, Size of message queue
  - Programming Best Practices
    - Retry with exponential backoff
- **How to react to incidents quickly?**
  - Monitoring (Logging, Tracing, Debugging, Metrics, Dashboards)
  - Being Prepared (Game days)
- **Post-mortem:**
  - Goal: How to prevent a repeat of the incident?
    - Goal is NOT "Whom to blame?"

Architect Responsibilities: Data Management

In 28 Minutes

- **Managing data and its flow** is a very important responsibility
  - **How** does data come in? (stream or batch or transactional application)
    - If you have plans to move from batch to stream, prefer Cloud Dataflow
  - **What** rate will we receive data?
    - Plan for scaling storage (prefer auto scaling storage like Cloud Storage)
  - **What kind of data?** (archive or ..)
  - **How much data?** (GB, TB or PB)
  - **For how long?**
    - Cloud Storage Lifecycle Policies and Retention Policies
    - Table Expiration
  - **Who will have access?**
  - **How will you use the data?**
    - Ideally keep data and data processing system in the same zone (or atleast the same region)
    - If you are not planning to use the data for a long time, use Cloud Storage Archive Storage class