Cloud Intro Notes

Imagine you have a cutting-edge idea in 2023 and need to build a solution using modern technology. In a traditional IT environment, you might request three separate servers—one for the web server, one for the application server, and one for the database—following a classic three-tier architecture. This process involves multiple steps and dependencies:

1. Submit a hardware order to your IT department.
2. Arrange for essential requirements such as power, cooling, rack space, and physical security.
3. Coordinate with your operational and security teams to install the necessary software.
4. Receive access to set up and run your new technology solution.

This entire process can take anywhere from several days to months, depending on approval processes and hardware availability.

In this traditional model, you are responsible for managing multiple layers of infrastructure, including:

- Networking and Internet access
- Storage and security (physical and access control)
- Server hardware and applications
- Database installation and configuration
- Governance, compliance, patches, migrations, upgrades, and environmental considerations (cooling and power)

While this approach provides increased security, customization, and control through direct management of every system facet, it also comes with significant drawbacks. The overall cost is high relative to the value, and scaling is not immediate. For example, doubling the number of servers—from three to six—may incur delays of days, weeks, or even months.

Cloud Computing: Definition and Operation

Cloud computing offers IT resources on demand, including compute power, application hosting, database services, and networking. It uses an API-driven model, meaning resources can be provisioned automatically when a client sends a request—whether through a website, command-line interface, or API.

For example, if you need a MySQL database with one terabyte of storage, four CPUs, and 32 GB of RAM, cloud providers like GCP automatically configure and deploy these resources. Instead of investing in physical hardware that is hard to return or repurpose, you only pay for the resources you use over the exact period they are active.

Cloud Deployment Models

Cloud consumption generally falls into three primary models:

1. **All-in-Cloud:**
   Organizations opt to run all workloads on cloud providers like GCP. This model is popular among startups and modern enterprises because it enables scalable, on-demand resource allocation without the need to manage physical hardware.
2. **On-Premises (Private Cloud):**
   Organizations maintain their own data centers and use virtualization technologies to manage resources internally. This model requires complete oversight of the entire stack—from hardware to software—and is often selected by companies with strict security and compliance mandates.
3. **Hybrid Cloud:**
   This model combines cloud and on-premises resources. Some workloads run in the cloud, while others remain in traditional data centers. Hybrid clouds are designed to leverage the best of both worlds, usually connected through high-speed links. It is important to note that hybrid cloud differs from multi-cloud, which involves using services from multiple cloud providers.

GCP vs. Traditional IT

Imagine building your own kitchen to make a pizza from scratch—you must gather ingredients, buy equipment, and manage every detail. This is similar to setting up a traditional IT environment where you purchase and configure hardware such as servers, network cables, power, and cooling. Alternatively, ordering a pizza from a restaurant like Pizza Hut saves you the hassle. Similarly, GCP takes care of the underlying operations for you. With GCP, you simply use ready-to-go services rather than managing the physical infrastructure yourself.

When using a conventional data center, you need to request server hardware, coordinate with operations and security teams, and wait days or even months for access. Consider the following diagram that compares these traditional IT setup requirements:

In contrast, GCP automates many of these tasks. Here's what changes when you choose GCP:

- You interface with pre-built services that include networking, cooling, and power.
- You access virtual machine services, database services, and application services that are ready to configure.

Ways to Interact with GCP

There are three primary methods to interact with GCP:

1. **GCP Console**
   The GCP Console is a web-based interface that offers an intuitive way to explore and create resources. It's ideal for beginners and for validating your interactions with GCP services.
2. **Cloud SDK (gcloud CLI)**
   The GCP CLI lets you execute commands directly in your terminal to retrieve information or manage resource
3. **Cloud Shell**
   Accessible directly in the GCP Console.
4. **Terraform (Provisioning GCP infrastructure declaratively)**

Benefits of Cloud

1. Converting Capital Expenditure (CAPEX) into Operating Expense (OPEX)

One of the major advantages of utilizing GCP is the ability to transform substantial upfront hardware investments into manageable monthly operating expenses. Rather than investing heavily in physical servers and infrastructure, you can use GCP to acquire virtual machines on a pay-as-you-go basis. This means you're billed only for the resources you actually use and can efficiently scale down when those resources are no longer required.

Upfront expense: CAPEX

Variable Expense OPEX

2. **Moving Away from Data Center Dependence**

A significant benefit of GCP is that it alleviates the need to manage your own data centers. For many organizations, such as insurance companies, running a data center is not a core competency. By shifting these responsibilities to GCP, you can concentrate on your primary business objectives, such as customer service and innovative application development.

3. **Enhanced Scalability on Demand**

Traditional data centers are often constrained by fixed hardware capacities and limited vendor supply, forcing companies to anticipate future requirements. GCP offers dynamic scalability, allowing you to rapidly adjust your virtual machines or database resources as demand fluctuates. This flexibility reduces risks associated with over- or under-provisioning.

Stop Guessing Capacity

4. **Leveraging Economies of Scale**

GCP capitalizes on massive economies of scale, meaning that as your consumption of cloud services increases, the cost per unit decreases. For instance, while the initial cost per gigabyte might be higher, high-volume usage can lead to significantly reduced rates. GCP's continual push to lower prices on services like Amazon S3 further enhances cost savings and budget predictability.

Cloud Design Principles:

1. Designing for Failure
2. Decoupling Components
3. Implementing Elasticity
4. Thinking in Parallel

Designing for Failure

When building any system, planning for failure is crucial. Consider a car: if one out of four wheels fails and the car stops functioning, that single point of failure can be catastrophic. Similarly, in cloud systems, the failure of a single component should not compromise the entire system.

To mitigate such risks, we design systems with redundancy and ensure they can automatically recover from failures. Embracing the philosophy that "everything fails all the time" (as noted by Werner Vogels) encourages us to build resilient systems that assume failure and plan robust recovery strategies.

Decoupling Components

The second principle is to decouple system components so that a failure in one part does not affect others. In tightly coupled architectures, a single fault can trigger a cascade of failures. In contrast, a loosely coupled system uses techniques such as queuing mechanisms and independent scaling, ensuring that each component operates in isolation from failures in connected parts.

For instance, if a front-end web server receives a flood of customer requests, decoupling the back end using a queue allows the server to manage the load at its own pace. This approach helps prevent data loss and maintains system integrity, especially under variable loads.

Implementing Elasticity

Elasticity is one of the standout advantages of GCP. Traditional data centers require significant time and resources to scale capacity up or down. GCP, however, enables you to automatically adjust resource allocation based on demand. When additional computing power is needed, GCP can quickly provision more resources and release them once demand decreases.

This elasticity improves performance during traffic surges while also optimizing costs since you only pay for what you use. According to GCP documentation, elasticity involves the automated acquisition and release of resources, ensuring efficiency and cost-effectiveness

Thinking in Parallel

The final principle is to embrace parallel processing rather than a strictly sequential approach. In a serial processing model, a prolonged task on a single server can become impractical. By contrast, parallel processing distributes tasks across multiple servers, drastically reducing processing time.

For example, a task that would take 36 hours on a single server can be divided among three servers to finish in 12 hours—or even among 36 servers to complete it in about 1 hour. GCP's ability to rapidly scale instances makes parallel processing a highly effective method for managing large tasks efficiently.