***ASSOCIATE CLOUD ENGINEER***

***IAM Policies***

Assign members(users) to roles (combinations of roles). WHO can do WHAT, on WHICH resources.

1. Basic roles –

Role Permission

Viewer View/ read- only (no modification allowed)

Editor Viewer + Modify existing data

Owner Viewer + Editor permissions w/ Manage roles & Permissions for project it’s resources & setting up billing for a project.

1. Predefined roles provide granular access for a specific service.
2. Custom roles provide granular access according to a user-specified list of permissions.

***Resource Hierarchy***

Organization -> Folder -> Project -> Resources.

When you give a user, group, service account a role on a specific element of the resource hierarchy, the resulting policy applies to the element you chose, as well as to elements below it in the hierarchy.

***Google Cloud solutions for various Consoles***

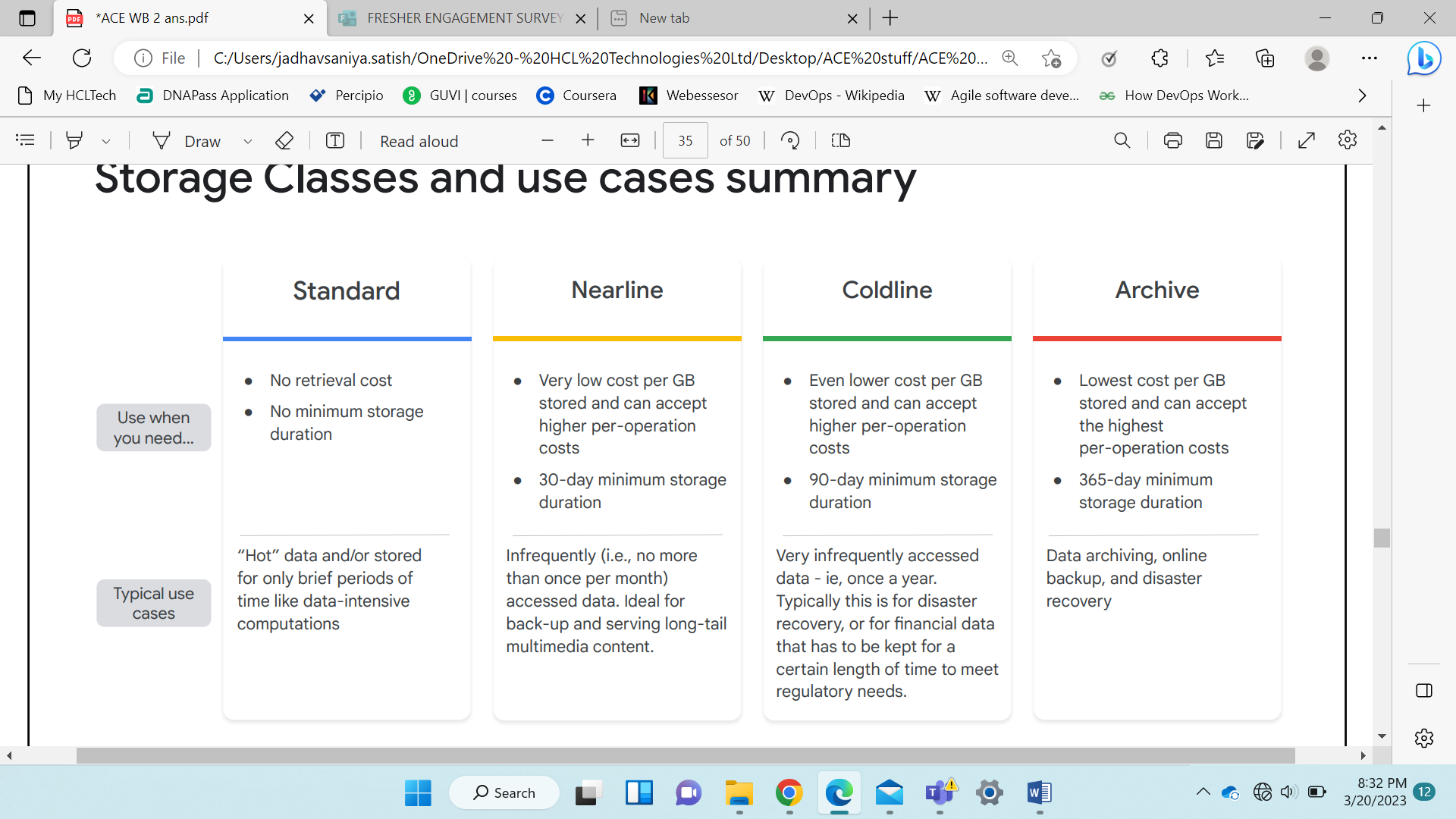
* **Google Cloud Console** – Web User Interface
* **Cloud SDK & Cloud Shell** – Command-line Interface
* **Cloud Mobile App** – For iOS & Android
* **REST-based API** – for Custom Applications

***PLANNING & CONFIGURING***

Plan & Estimate price of GC using calculator.

P&C network resources, compute resources, device storage.

* **Compute options** in Google Cloud are those based on virtual machines as well as those based on containers.
* **Standard storage** – frequently in use.
* **Nearline storage** - data accessed every 30 days.
* **Coldline storage** - once per quarter (90 days).
* **Archive -** less than once a year.



* **Ingress** (data writes) is free.
* **Egress** (amount of data read from the bucket) is chargeable.
* **Google Kubernetes Engine** is PAAS for container management. User has Full control on container orchestration & Google manages control plane. You don’t manage specific operating system dependencies. Logical based & manages containers in production. Server-based.
* Containerized workloads. Cloud-native distributed systems. Hybrid applications.
* **Compute Engine** gives you full control over operating system choice and configuration (user can do OS level changes). Implement & manage VM that apps run on (in Google’s Global data centers) able to move to the cloud without rewriting your code. Server-based.
* Run on-premises s/w on cloud. Any workload which require specific OS.
* (group, subgroup, command - list, describe & delete) CE disk snapshots – gcloud compute snapshots list --project PROJECT\_ID
* To describe snapshots- gcloud compute snapshots describe SNAPSHOT\_NAME
* **App Engine** is PAAS for code deployment on infrastructure managed by Google. 2 states – Standard has abstract & Flexible has many options for deployment. You don’t manage operating system dependencies. Serverless.

Build Apps. Focus on writing code, Developer velocity, minimize operational overhead.

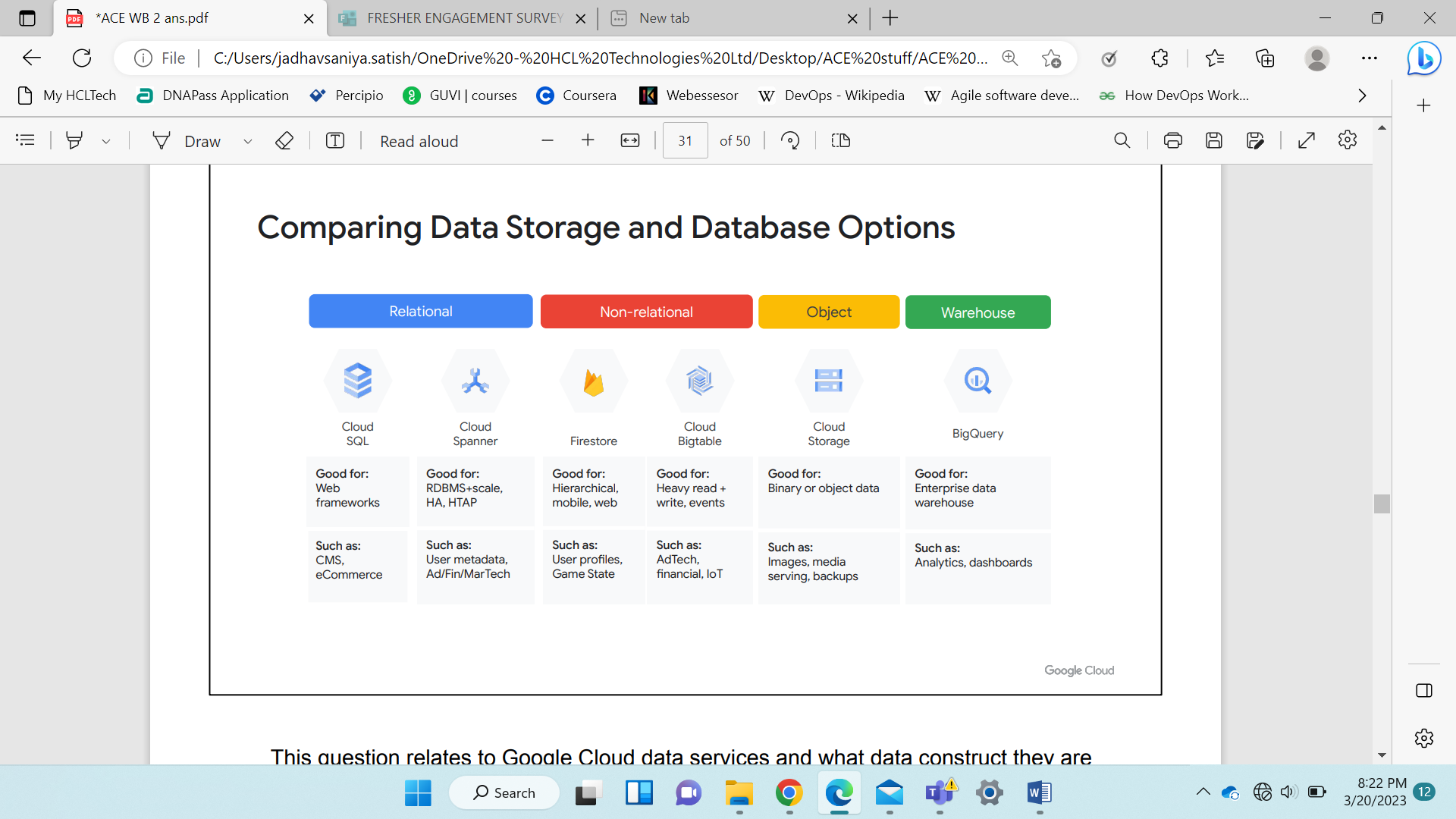
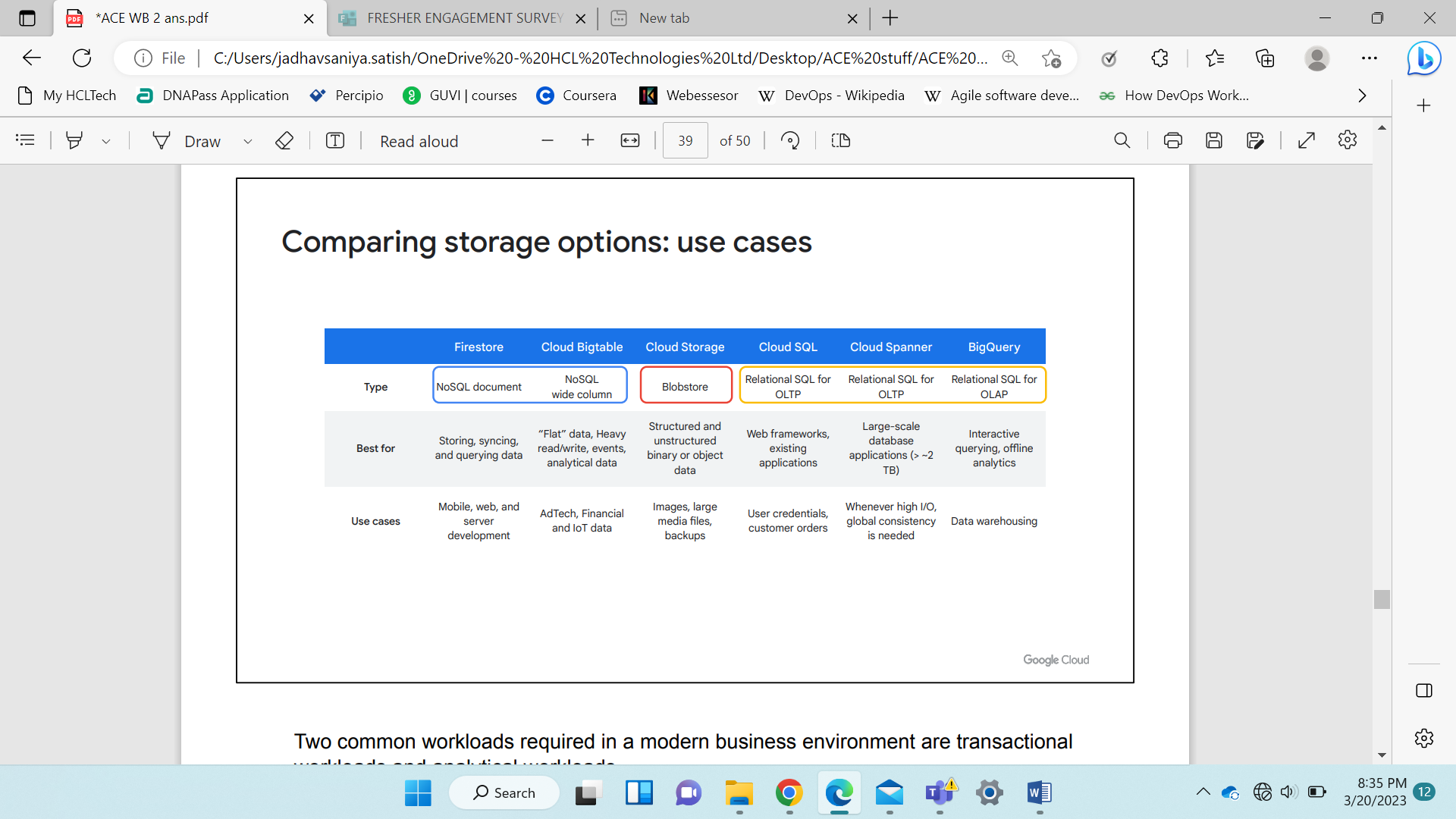
* Gaming back-ends, apps, websites, IOT applications.
* **Cloud Run** deploys containers/code in Google Cloud which listens requests from web/events of GC. User can’t specify cluster & deployment architecture. Serverless.

Support API end-points. Pay for use only. Scales on-demand.

* Web frameworks, micro services.
* **Cloud Functions** – Execution environment. Serverless. Build & execute cloud services. Manages code as short/ executable functions rather than containers.

Minimal config. Scales on-demand. Event driven workloads.

* Statistical analysis, Image thumbnail generation, Posting comment to Slack channel after GitHub commit.
* **Data services with modern data constructs** – relational(transactional), non-relational/no-sql, object/ warehouse (historical & analytical). For **cloud storage classes** - regional, multiregional, nearline and coldline.
* **Cloud Spanner** - SQL-compatible relational DB. Horizontally scalable and globally available.
* **Cloud SQL** – relational DB optimizes transactional read/writes. Manages/implements on-premise.
* **BigQuery** – Modern warehouse having historical & SQL query engine. serverless distributed query engine. No native but query external data.
* **Cloud Firestorm** - NoSQL document DB define entites & attributes.
* **Cloud Bigtable** – NoSQL based on columns where rows are referenced by key that combines common queried columns. store historic, time-based data.

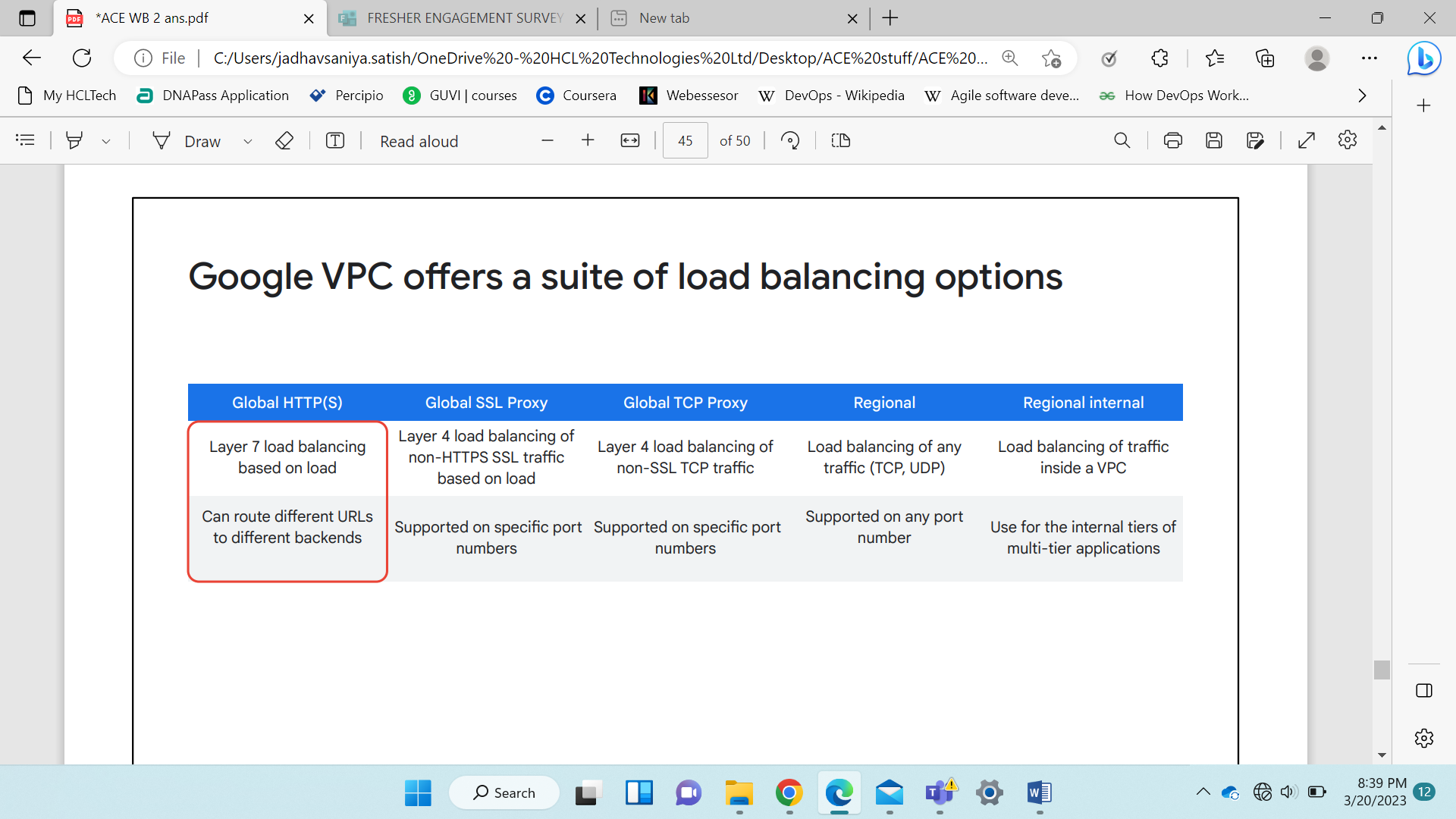
 

***Load balancing -*** designed for applications that is used by end users so that LB can perform the application well when it faces spike in user activity.

* **mapping w/ TCP layer**- OSI model L7 is Application Layer. **http(s), ftp.**

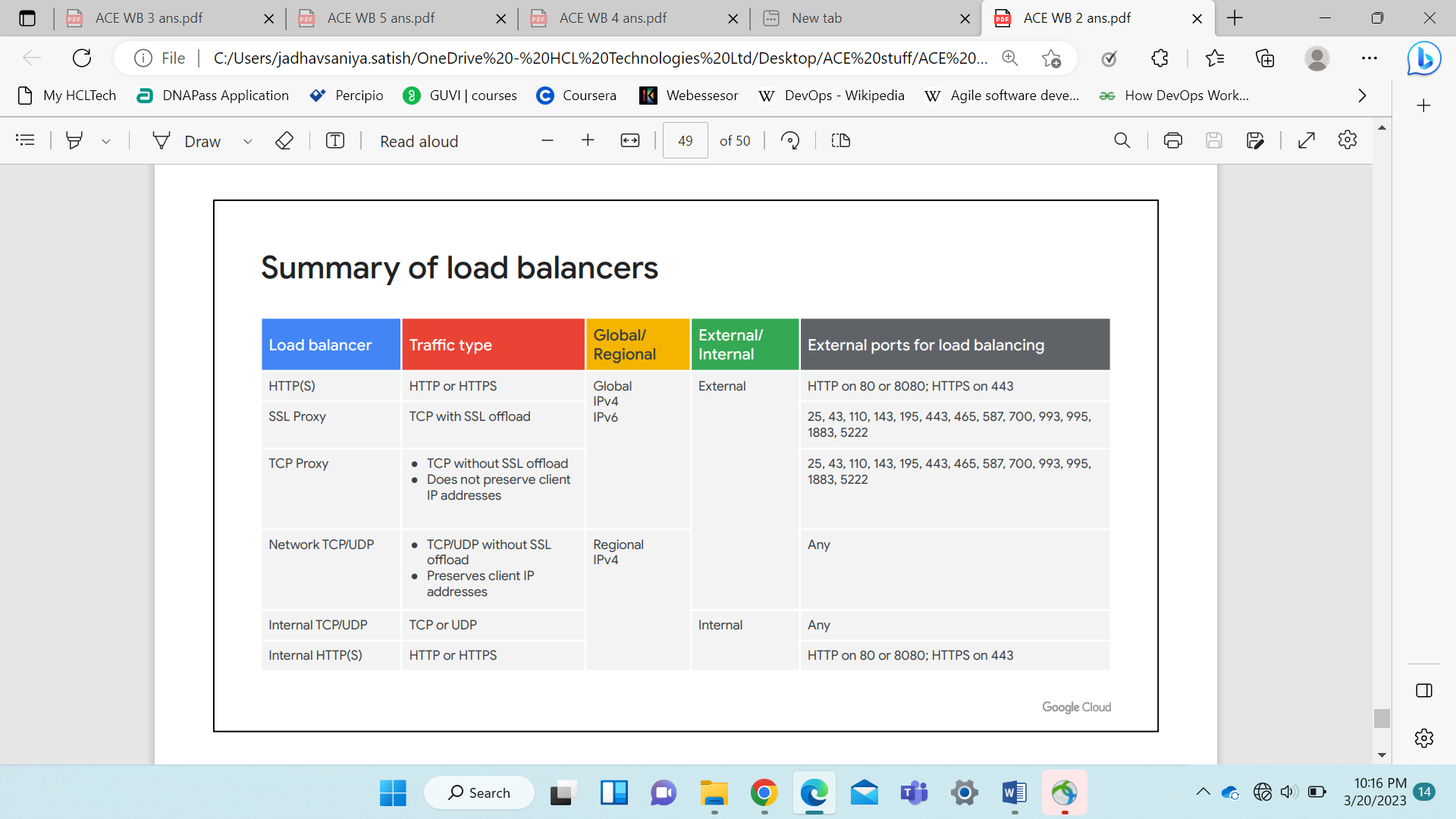
L4 is Transport n Network L (host2host comm.) **TCP/UDP, SSL/Global SSL Proxy, Global TCP Proxy, Regional Network/ network LB.**

* **Internal & external LB** - Internal include TCP/UDP, http(s), and network pass-through LB. External LB https, SSL, and TCP load balancers.
* **Proxied LB** - terminates the connections and proxy them as internal new connections.
* **Pass-Through** - LB passes the connections directly to backends.



* HTTPS - https LB is proxied. Recommended to use for web frontends. Premium https is costly.
* Regional Internal LB - doesn’t require external IPs and is more secure.
* TCP/UDP is a pass-through balancer.
* SSL - Premium tier SSL is global.

Scenario - When load balancing in a particular region, external connectivity to your front ends can happen through an external http(s) load balancer with the proper forwarding rules and the standard networking tier. For connectivity internal to your defined vpc network, you should use the internal https and internal TCP/UDP load balancing options.



Which services are based on logic implemented in containers

Ans- 1) Google Kubernetes managed Kubernetes environment that lets you deploy containerized apps via pods, deployments, and services you specify.

2) CLOUD RUN-  Cloud Run is a serverless offering that runs your containerized code when monitored events take place.

Which data storage service is a unique globally available, horizontally scalable database with relational semantics?

Ans- CLOUD SPANNER

***PROFESSIONAL CLOUD SECURITY ENGINEER***

***JOB ROLE***

1. Secure workloads and infrastructure to be designed.
2. Having knowledge and keeping in mind industry and security best practices it should be design, develop, maintain by using(leveraging) Google Cloud Techniques.
3. 3+ Industry Experience and 1+ Experience in designing and managing GC services and tools.

***Proficient in***

1. Identity and Access Management.
2. Use Goggle Cloud for data privacy.
3. Configuring network security defenses.
4. Gathering Goggle logs.
5. Manage Incidents.
6. Define organization’s structure and policies.
7. Apply dynamic regulatory considerations.

* **Hybrid Cloud Model** - On-premise when connected to G Cloud.
* **Microsoft Active Directory/ LDAP**
* **GCDS (Directory Sync)** – tool to sync on-premises Active directory users+ groups to Cloud Identity Domain
* **Security Assertion Markup Language (SAML) - SAML2** is identity provider & GC is service provider
* Corporate active directory SAML2 for SSO (Single Sign-on)

***Secure network resources***

* **Design network security** – keep traffic private/ limited access to internet so attackers won’t get endpoints to break into the system.
* **Network segmentation** – protect resources by configuring website application firewall Cloud Armor. Establish patterns – to monitor invalid/ attack traffic/ block/filter it. establish private connectivity- leverage isolation and segmentation to restrict type of communication occurring internally. Ports & protocols used for standard workloads are not exposed to public internet & ad actors.
* **HTTPS Load Balancers** – protect layer 3 + layer 4 (network & transport) DDOS defense.

***Google Cloud Armor*** provides a configurable managed service integrated with the HTTPS LB and protects against layer 7 (application layer) attacks.

**Protection against** - SQL injection, cross-site scripting injection, and similar attacks.

**Filter traffic / throttled** with configurable rate limits or challenged with a reCAPTCHA for bot management or to further protect against DDoS.

**Identity-Aware Proxy or IAP** - If public n/w used to connect via HTTPS or any TCP protocol (SSH) requires authenticated and authorized access hence they will connect through IAP. Use Google-managed identities with Google authentication or SAML 2-federated single sign-on or any external identities and other types of single sign-on when combining the IAP with Identity platform.

**DNSSEC** – By activating it secures for public DNS zones to prevent attackers from manipulating DNS responses to DNS requests sent to these public zones.

Cloud DNS ensures authenticated DNS responses to DNS requests and automatically manages DNSSEC-related DNS records. (DNSSEC is supported by Cloud DNS and also by Google Domains, Google's domain registration and registry service.)

**Firewall** rules – allow valid and expected network traffic between IP addresses, ports,& protocols used by actual workloads. Block all other traffic. Stateful rules handle requests in either the ingress or egress direction. Wherever possible, define firewall rules to restrict traffic to specific source and target service accounts. Use hierarchical firewall rules to ensure uniform application of rules across projects.

Use of private IP address communication between all resources within GC happen primarily inside shared VPC networks. Shared VPC provides centralized network administration and allows resources from a large number of projects to communicate via a single VPC. Isolated standalone networks can be VPC peered to the shared VPC networks if they need to communicate with standard workloads running in the shared VPC networks. VPC peering provides decentralized network administration and allows VPCs to be connected across projects or even organizations.

Cloud VPN and Interconnect provide private IP communication from on-premise networks to Google Cloud. Utilize private IP communication from office and data center networks into GC via Cloud VPN and Interconnect. Google private access enables private IP communication to Google APIs.

Cloud NAT allows resources with only internal IP addresses to make requests to the public internet. All internet-bound requests through Cloud NAT, allowing them to only be given internal IP addresses.

