**Question A**

|  |  |  |  |
| --- | --- | --- | --- |
| **Traditional On-Premises** | | **Modern Cloud** | |
| **On-Premises Definition** | **On-Premises Example** | **Cloud Definition** | **Cloud Example** |
| **Monolithic:**  In information technology, it could be referred as composed all in one piece or very big, depending on the context used (Wigmore, 2016). It is used in many ways to describe integrated circuits, applications, organizations and storage systems (Wigmore, 2016). | **Monolithic:**  -Monolithic Architecture (Singular Monolith, Distributed Monolith, Third-party Monolith) (Codurance, n.d.).  -Monolithic Integrated Circuit (IC) (Wigmore, 2016).  - Monolithic Storage Array (Wigmore, 2016).  - Monolithic Corporation (Wigmore, 2016). | **Decomposed:**  Is one of the four parts of computer science, it is the breaking down of complicated problems or system into smaller manageable and easy to understand parts. The parts being smaller makes it easier to exam, solve and work on the problem (Bitesize, n.d.). | **Decomposed:**  -A good example would be the creation of a website as the front-end developers could each work on a few pages of the website, which they all post the work they have done on a repo on GitHub which the project leader/manager will merge all the parts into one and should there be problems with the login page during the merger, the problem will be easily spotted and be fixed without the other part of the project being affected (inspired by: (Tarunsinghwap7, 2022)) . |
| **Designed for predictable scalability:**  The measure in ability of a system to decrease or increase in cost and performance in reaction to changes in system processing and application demands (Gartner, n.d.). | **Designed for predictable scalability:**  -How well the website performs when a lot of users trying booking flights when plane tickets are on sale (Gartner, n.d.). | **Designed for elastic scale:**  The ability for a system to manage available resources corresponding to the present workload requirements dynamically (The App Solutions, n.d.). | **Designed for elastic scale:**  -Better fault tolerance (In AWS environments, elastic scale can spot when a server is unhealthy, stop it and start an instance to replace it) (AVI Networks, n.d.)  -Better Availability (AVI Networks, n.d.).  -Better Cost Management (AVI Networks, n.d.). |
| **Relational database:**  A type of database which keeps and supply access to data points which are associated to each other (Oracle, n.d.). | **Relational database:**  -Oracle Database (Database Town, n.d.).  -Microsoft SQL Server (Database Town, n.d.)  -IBM DB2 (Database Town, n.d.).  -MySQL (Database Town, n.d.). | **Polyglot persistence (mix of storage technologies):**  It basically means the use of different data storage technologies to hold different data storage needs (Object Rocket Marketing, 2018). It came from polyglot programming which is the use of different programming languages to bluid an application (Object Rocket Marketing, 2018). | **Polyglot persistence (mix of storage technologies):**  -Netflix (makes use of relation, columnar, document and key-value data stores for the storage of ratings, profile, recommendations, subtitles, videos and metadata. That is how the personalized experience is delivered to all millions of users.) (Al & LinkedIn Community, n.d.).  -LinkedIn (Al & LinkedIn Community, n.d.).  -Amazon (Al & LinkedIn Community, n.d.). |
| **Synchronized processing:**  The coordination of execution of many processes in a multi-process system to ensure that shared resources are obtained in an expected and controlled manner (GeeksforGeeks, 2023) | **Synchronized processing:**  -User Interfaces (Interaction between humans and computers, as humans expect to get response immediately as that the communication standards familiar to humans) (Broshar, 2021).  -HTTP APIs (Quick answer from the web server is expected from Client programs sending HTTP requests) (Broshar, 2021). | **Asynchronized processing:**  The plan of computing tasks in a way which the tasks can be executed without them having to depend on each other (Spacey, 2023). The request response is not waited for by the client when sending a request as the response could take a few minutes or hours or days because the response is not necessarily sent back. Only the confirmation of the request being received is sent (Tarnowski, Adrian, n.d.). | **Asynchronized processing:**  -Long-running Tasks (Orders on e-commerce sites are done asynchronously so there’s no unnecessary resource blocking.) (Broshar, 2021).  -Batch-processing (Huge amounts of data handling data asynchronously method is known as data-processing. These huge batches of data are processed at scheduled times so computer resources blocking is avoided.) (Broshar, 2021). |
| **Design to avoid failures (MTBF):**  The system’s average time between breakdowns. It is a crucial maintenance metric used to measure safety, performance and equipment design mostly for assets found to be complex like airplanes (Fiix Software, n.d.). | **Design to avoid failures (MTBF):**  For example, the certain asset has been operational for 2500 hours in a year and it broke down 10 times that year. So, the MTBF will be calculated as: *MTBF = # of operational hours ÷ # of failures*  MTBF = 2500 ÷ 100  MTBF = 250 hours  (Fiix Software, n.d.) | **Design to for failures (MTTR):**  The maintenance metric used to measure the average time required to repair and troubleshoot failed equipment. The response time to unplanned breakdowns being repaired (Fiix Software, n.d.). | **Design to for failures (MTTR):**  For example, if 82 hours were spent on fixing an unplanned breakdown of an asset which breakdown 12 times in a year, the calculation of MTTR would be:  *MTTR = Total maintenance time ÷ Number of repairs*  MTTR = 82 ÷ 12  MTTR = 6.83 hours.  (Fiix Software, n.d.) |
| **Occasional large updates:**  Bringing a system/feature up to date by incorporating new features/information or rectifying a large portion of the system/feature after a periodic time (Dictionary, n.d.). | **Occasional large updates:**  -Taking a month/s or a year to incorporate new features into the system (Dictionary, n.d.). | **Frequent small updates:**  Bringing a system/feature up to date by incorporating new features/information or rectifying a small portion of the system/feature very often (Dictionary, n.d.). | **Frequent small updates:**  -Taking a week, bi-week or latest three weeks to incorporate new features into the system (Dictionary, n.d.). |
| **Manual management:**  The system/management where maintenance and/or management are done by hand without the use of any automatic system and/or computing system (PadaKuu, 2023). | **Manual management:**  -For example, the administrators having to spend too much time searching for documents which were misplaced which cost the organisation money and time (inspired by (PadaKuu, 2023)). | **Automated self-management:**  The combination of both hardware and software that are designed and programmed to automatically work without the need of human interaction/operator to give inputs and/or instruction for the operations to be executed (PadaKuu, 2023). | **Automated self-management:**  -For example, the bio-entrance system which automatically update the file when an employee clocks in/out at work without the employee having to report to the reception when clocking in or out (inspired by (PadaKuu, 2023)). |
| **Snowflake servers:**  Servers which run a mission critical bit of software that only runs on a certain specific configuration of application server and operating system as the cannot be upgraded (Progress Kemp, n.d.). | **Snowflake servers:**  -The server is destroyed completely so that a new one is used to replace it when changes are made (Zislis, 2017).  -The servers support warehouse sizes which range from X-Small (1 credit/hours) to 6X-Large (512 credits/hours) (Snowflake, n.d.). | **Immutable infrastructure:**  An infrastructure model whereby servers are never altered after being stationed (Vidró, 2017). | **Immutable infrastructure:**  -Containers (Continuous changes cannot be made to a made because a new version of the container needs to be made or a recreation of the container existing from its picture.) (Cloud Native Glossary, 2022) |

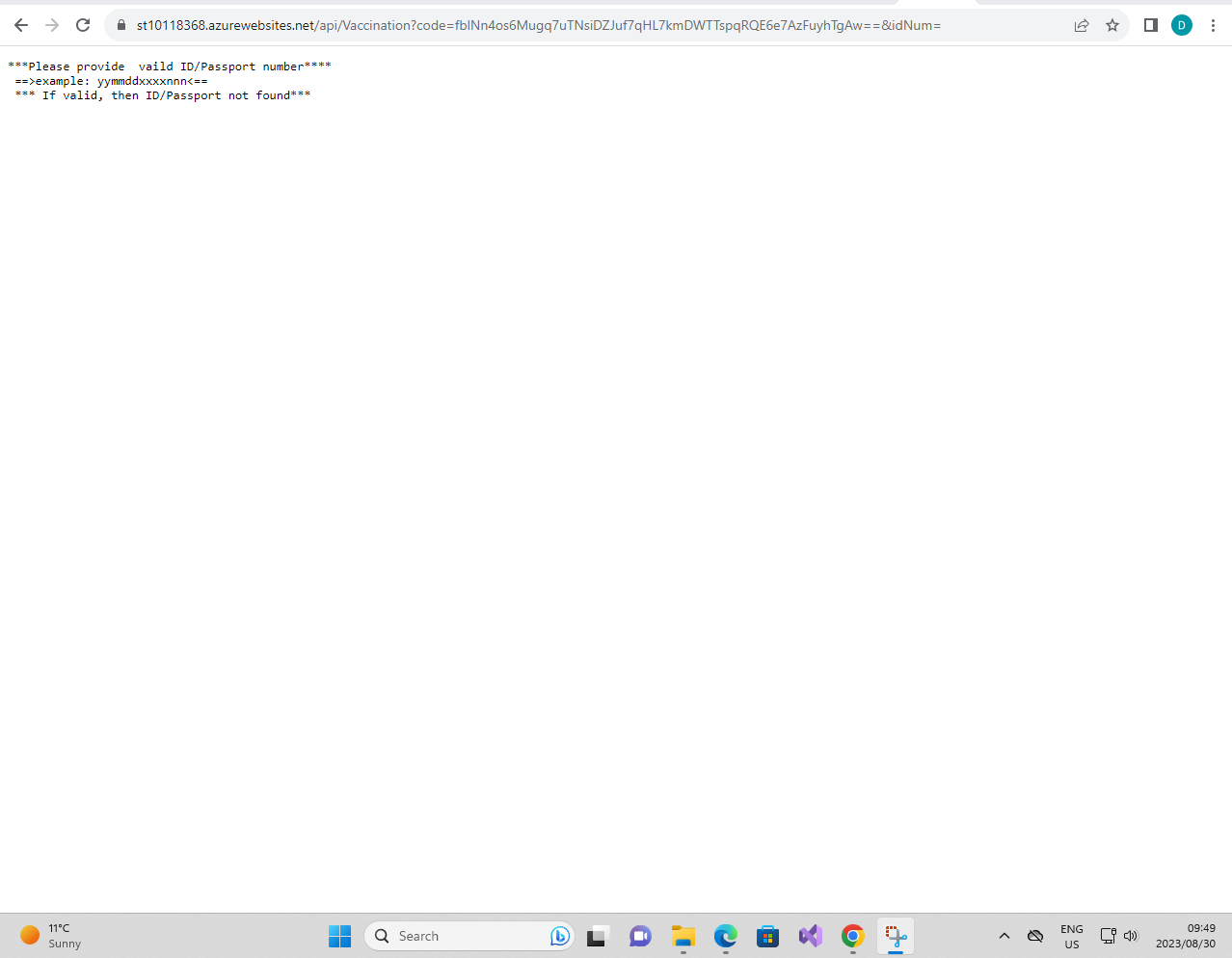
**Question B**

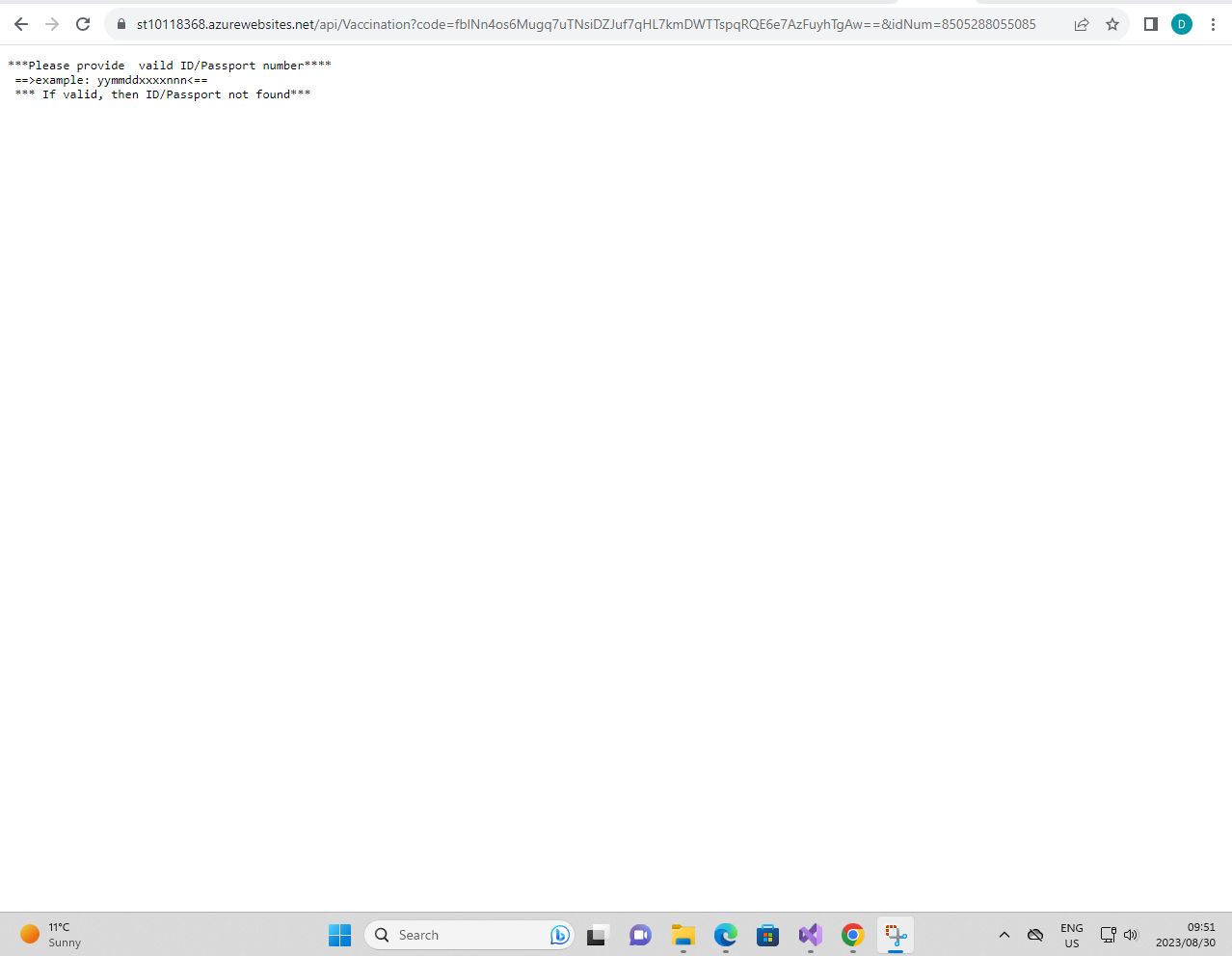
**Function Code:** A computer screen with text on it

Description automatically generatedA computer screen shot of a program

Description automatically generated

**Function on Web Browser:**





A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

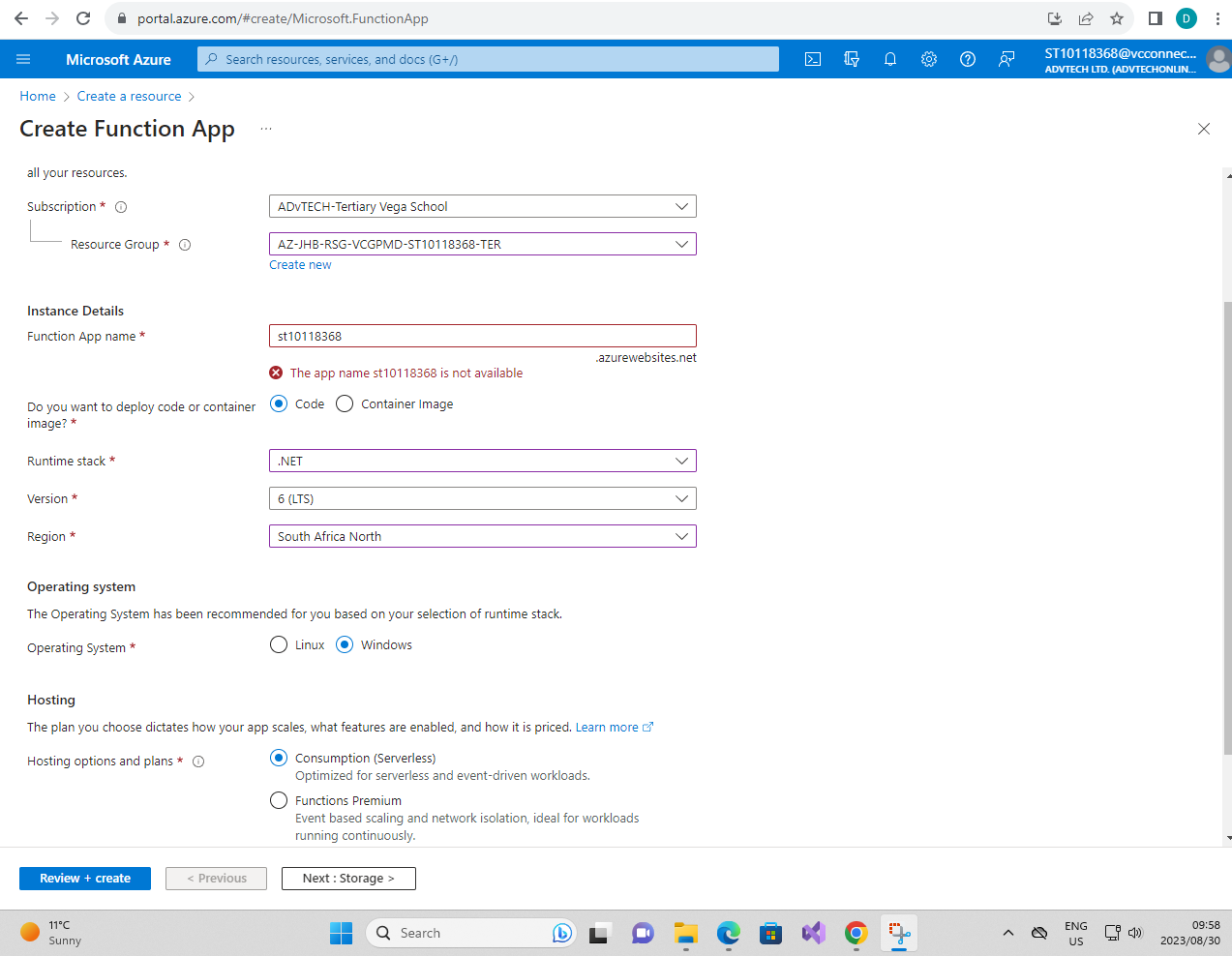
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Deployment:**



A computer screen shot of a computer

Description automatically generated

A computer screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen with a pink box

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

A computer screen shot of a computer screen

Description automatically generated

**==NOTES==**

**Live Link:** [https://st10118368.azurewebsites.net/api/Vaccination?code=fblNn4os6Mugq7uTNsiDZJuf7qHL7kmDWTTspqRQE6e7AzFuyhTgAw==&idNum=\*\*\*\*\*\*](https://st10118368.azurewebsites.net/api/Vaccination?code=fblNn4os6Mugq7uTNsiDZJuf7qHL7kmDWTTspqRQE6e7AzFuyhTgAw==&idNum=******)\*\*\*\*\*\*\*

Swap \*\*\*\*\*\*\*\*\*\*\*\*\* with ID/Passport Number

**ID/Passport Numbers:**

8505288055081

0501014816083

0101014815083

9810125015082

# References

Al, & LinkedIn Community. (n.d.). *How can you use the polyglot persistence pattern effictively?* Retrieved from LinkedIn: https://www.linkedin.com/advice/1/how-can-you-use-polyglot-persistence-pattern

AVI Networks. (n.d.). *Elastic Scale*. Retrieved from AVI Networks: https://avinetworks.com/glossary/elastic-scale/#:~:text=Better%20fault%20tolerance%20%E2%80%93%20for%20example,an%20instance%20to%20replace%20it.&text=Better%20availability%20%E2%80%93%20elastic%20scaling%20helps,handle%20the%20current%20traffic%20demand.

Bitesize. (n.d.). *Decomposition*. Retrieved from Bitesize: https://www.bbc.co.uk/bitesize/guides/zqqfyrd/revision/1#:~:text=Decomposition%20is%20one%20of%20the,are%20simpler%20to%20work%20with.

Broshar, A. (2021, March 15). *Introduction to Synchronus and Asynchronous Processing*. Retrieved from Koyeb: https://www.koyeb.com/blog/introduction-to-synchronous-and-asynchronous-processing

Cloud Native Glossary. (2022, November 09). *Immutable Infrastructure*. Retrieved from Cloud Native Glossary: https://glossary.cncf.io/immutable-infrastructure/#:~:text=Containers%20are%20a%20good%20example,existing%20container%20from%20its%20image.

Codurance. (n.d.). *Monolith and Coupling Types*. Retrieved from Codurance: https://www.codurance.com/publications/types-of-monolith-coupling#:~:text=Monolithic%20architectures%20have%20been%20a,%2C%20and%20Third%2Dparty%20Monolith.

Database Town. (n.d.). *13 Examples of Relational Database*. Retrieved from Database Town: https://databasetown.com/examples-of-relational-database/

Dictionary. (n.d.). *update*. Retrieved from Dictionary: https://www.dictionary.com/browse/update

Fiix Software. (n.d.). *Mean time between failures*. Retrieved from Fiix Software: https://fiixsoftware.com/maintenance-metrics/mean-time-between-fail-maintenance/#:~:text=Mean%20time%20between%20failures%20(MTBF)%20is%20the%20average%20time%20between,assets%20like%20generators%20or%20airplanes.

Fiix Software. (n.d.). *Mean time to repair (MTTR)*. Retrieved from Fiix Software: https://fiixsoftware.com/maintenance-metrics/mean-time-to-repair-maintenance/

Gartner. (n.d.). *Gartner Clossary*. Retrieved from Gartner: https://www.gartner.com/en/information-technology/glossary/scalability

GeeksforGeeks. (2023, 02 01). *Introduction of Process Synchronization*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/introduction-of-process-synchronization/

Object Rocket Marketing. (2018, 04 30). *What is Polyglot Persistence?* Retrieved from Object Rocket: https://www.objectrocket.com/blog/uncategorized/what-is-polyglot-persistence/

Oracle. (n.d.). *What is a Relational Databse (RDBMS)?* Retrieved from Oracle: https://www.oracle.com/za/database/what-is-a-relational-database/#:~:text=A%20relational%20database%20is%20a,of%20representing%20data%20in%20tables.

PadaKuu. (2023, May 06). *Difference Between Manual And Automated System - Manual System vs Automated System*. Retrieved from PadaKuu: https://padakuu.com/difference-between-manual-and-automated-system-manual-system-vs-automated-system-1-article

Progress Kemp. (n.d.). *Snowflake Servers*. Retrieved from Progress Kemp: https://kemptechnologies.com/resources/glossary/snowflake-servers#:~:text=Snowflake%20servers%20run%20a%20mission,operating%20system%20and%20application%20server.

Snowflake. (n.d.). *Overview of Warehouses*. Retrieved from Snowflake: https://docs.snowflake.com/en/user-guide/warehouses-overview

Spacey, J. (2023, January 15). *What is Asynchronous Processing?* Retrieved from Simplicable: https://simplicable.com/design/asynchronous-processing

Tarnowski, Adrian. (n.d.). *Synchronous vs. asynchronous communication (in the cloud and beyond)*. Retrieved from Predica: https://www.predicagroup.com/blog/synchronous-vs-asynchronous-communication/#:~:text=In%20case%20of%20the%20asynchronous,was%20received%20by%20the%20server.

Tarunsinghwap7. (2022, 01). *What is Decomposition Computational Thinking?* Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/what-is-decomposition-computational-thinking/

The App Solutions. (n.d.). *Cloud Elasticity Vs. Scalability: Main Differences To Know About*. Retrieved from The App Solutions: https://theappsolutions.com/blog/cloud/cloud-computing-elasticity-vs-scalability/#contents\_0

Vidró, H. (2017, September 26). *What Is Immutable Infrastructure?* Retrieved from Digital Ocean: https://www.digitalocean.com/community/tutorials/what-is-immutable-infrastructure

Wigmore, I. (2016, 05). *Definition*. Retrieved from TechTarget: https://www.techtarget.com/whatis/definition/monolithic#:~:text=Monolithic%2C%20in%20information%20technology%2C%20means,storage%20systems%2C%20among%20other%20things.

Zislis, E. (2017, March 7). *What are the pro and cons of SnowFlakes Servers, Phoenix Servers and Immutable Servers?* Retrieved from DevOps: https://devops.stackexchange.com/questions/49/what-are-the-pro-and-cons-of-snowflakes-servers-phoenix-servers-and-immutable-s