Cairo University

Faculty of Computers and Artificial Intelligence

Innovation in Storage : Opportunities and Challenges.

Nariman Tarek Ahmed Zobaa (Id:11422018492132)
Mai Ahmed Hussein Ayad(Id:11422018498477)
Nashwa sowelam Mohamed(Id:11422018498481)
Eman Abed Elhameed Ahmed Amin(Id:11422018434975)

Supervisor:Dr Shaimaa M. Mohamed

August 2020

Acknowledgement

We would like to express our special thanks and gratitude to our doctor **Shaimaa Mohamed**, who gave us her aid and advice and supported us through out this project till its successfully done.

We would like also to thank our families, who have supported us through thick and thin and who have helped us finish this project and end this phase of our lives. You are the most precious in all our lives.

Special words from Nariman:

- I would like to express my deepest gratitude to my beloved company **"DELL-EMC"** that gave me the opportunity to join this masters program and add a master degree to my resume and scientific achievements.
- Special thanks and gratitude to my supervisors **Ahmed Hamdi** and **Tarek Hamdi**, who have supported me throughout the two years of my study. I appreciate your support, aid, and advice that led to this success and accomplishment.
- And most of all, I owe my deepest gratitude to my husband, **Omar Emara**, for his affection, encouragement, understanding and patience. He supported me without any complaint or regret that enabled me to complete my thesis. My thesis acknowledgement would be incomplete without thanking my baby-daughter, Kenzy, whose smiling face always made me happy and inspired me. Having her midway during my Masters was certainly not easy for me but she has made my life wonderful.

Contents

Nomenclature

ADE	Azure Disk Encryption
API	Application Programming Interface
ARPANET	Advanced Research Projects Agency Network
AWS	Amazon Web Services
BLOB	Binary Large Object
CAPEX	Capital Expenditure
CLI	Command Line
EC2	Elastic Compute Cloud
EU	European Union
FIFO	First In, First Out
GPv1	General Purpose Version 1
GPv2	General Purpose Version 2
GRS	Geo-Redundant Storage
GZRS	Geo-Zone-Redundant Storage
HDD	Standard Hard Disk Drives
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IaaS	Infrastructure as a Service

IO	Input/Output
LRS	Local Redundancy Storage
NAS	Network-Attached Storage
NIST	National Institute of Standard and Technology
O/S	Operating System
PaaS	Platform as a Service
RA	Read Access
RA/GRS	Read Access Geo-Redundant Storage
REST	Representational State Transfer
S3	Simple Storage Service
SCSI	Small Computer System Interface
SQL	Structured Query Language
SSD	Solid-State Drive
SSE	Server Side Encryption
URL	Uniform Resource Locator
UTC	Coordinated Universal Time

List of Figures

Abstract

In this constantly changing world of increasing complexity and scalability, the amount of data that needs to be stored nowadays is growing exponentially, which makes innovation in technology mandatory for serving business needs and creating new opportunities. There are three main technologies that storage world is based on; On-premises, Cloud-based, and Decentralized Storage Networks (DSN). Each one of these technologies has its own usage and can be implemented to serve specific applications.

In this research, each one of these storage technologies is being studied with its advantages and dis-advantages. Also, it gives a technically comparison between each technology associated with the business recommendations; as it is becoming more challenging for a business to decide what is the optimum solution to implement in order to achieve its objectives and profits.

1 Introduction

In the age of technology explosion which we are witnessing, the advancement of technologies like IoT, AI, etc. brings new innovative solutions and impacts in shaping the new economic model that makes these technologies no longer “nice to have” technologies but a necessity. The shared game-changer for these technologies and in turn businesses of all sizes that depend on it is the availability of data from every source imaginable – social media, sensors, business applications, and many more. These technologies handle a massive amount of structured, and unstructured data and sparked the world by showing how a fully interconnected smart world can make life much better. In AI for example according to Umbel [4] , “Using data from multiple sources, AI can build a store of knowledge that will ultimately enable accurate predictions about you as a consumer that are based not just on what you buy, but on how much time you spend in a particular part of a site or store, what you look at while you’re there, what you do buy compared with what you don’t – and a host of other bits of data that AI can synthesize and add to, ultimately getting to know you and what you want very, very well,” while IoT generates a massive amount of data retained in vertical silos. Forecasts suggest that by 2030 around 50 billion of these IoT devices will be in use around the world, creating a massive web of interconnected devices spanning everything from smartphones to kitchen appliances [5].

Analyzing these massive amounts of data to get the insights of the business trend is defined as ”Big Data”. Big Data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, analyze, manage, and process within a tolerable elapsed time. Big Data is defined by three characteristics known as 3Vs which are Volume(the size of data), Velocity(the speed of data processing), and Variety(the different types of data), these characteristics support the process of Big Data analysis. So due to the persistent enormous growth of data nowadays, storage systems become the cornerstone and essential skeleton for the organizations to store these huge amounts of data in reliable storage that have low latency and enough capacity to suffice operational data.

However, the enormous growth in data creates the evolution of storage technologies to serve endless business needs in both on-premises and on Cloud. However, It also showed that the current communication infrastructure aka.HTTP, or the HyperText Transfer Protocol, regardless of being a “good enough” system for most use cases already existing now, might not be “good enough” while we are entering a new era of data distribution with new challenges (e.g., hosting petabyte data sets, real-time media streams, commuting on massive amounts of data across organizations, preventing losing important files, etc.). Thus, a new model of storage with different infrastructure is needed to fit in these novel processes. Hence there have been many attempts to fit in the new internet revolution’s requirements.

In this project, we conduct an overview of three types of storage; on-premises, cloud-based, and decentralized storage networks. Furthermore, we also identify their challenges and potential business opportunities. To the best of our knowledge, this is the first project that combines the three models and merges the technical review with business benefits. We anticipate that this work can shed new light on the subsequent studies related to storage models.

- **In Storage on premises:**We are discussing the available options of on-premises Storage architecture; Server-centric Architecture and Information-centric Architecture. Also, we are recommending the best on-premises storage option for organizations based on different factors such as capacity, scalability, backup, and recovery. The chapter illustrates Modern Data Center Infrastructure types (e.g. virtual Infrastructure, Software-Defined Infrastructure, and orchestration) and techniques for data protection and optimization by defining various types of RAID. This chapter also introduces three products produced by Dell EMC; XtremIO X2, a block storage array product, and Isilon, a File-based storage array product and ECS, Object-based storage array product.

- **In Storage on cloud :**We are discussing the appearance of storage on Cloud that removes the heavy burden from organizations as building data centers will need capital expenditures and hiring staff with the appropriate expertise. Mentioning the benefits of migrating organizations elements to a cloud computing environment. These organizations can choose to host their services in different cloud service providers like AWS, Azure and Google Cloud, Azure is one of the top players in the cloud technology, so it was essential to illustrate all storage services offered by Azure such as Blob, file-share, tables, and queues.

- **In Decentralized Storage:** We are articulating the new concept which is, from the architecture perspective, the contrast of the other alternatives. We are illustrating the state of the art of this novel technology that brings new economic models, could be described as disruptive, and giving an overview of its roots and how it has shaped over the years. Two main novel implementations of decentralized storage networks are being discussed in depth; IPFS and Filecoin as two of the most successful implementations that are widely supported.

Since the goal of this project is to explore the different models of storage that currently exist and follow up on the updates that currently trending in the technical world in order to identify their challenges and potential business opportunities, we are illustrating our outputs as follow:

1. Articulate the pros and cons of each technology.
2. Held a technical comparison that illustrates each model technical features
3. Give recommendations based on some business use cases to help businesses make decisions.

1.1 Research Hypotheses

1. People have issues/fears with vendor lock-in on cloud and on-premises storage technologies.
2. Few people will be aware of a decentralized storage networks.
3. People will be willing to mine with their storage and make revenue.

1.2 Methodology

In order to grasp the different part of each model, we have tackled different approaches :

- **Exploring** white papers, docs as well as the academic papers that have addressed these technologies.
- **Taking** hands-on experience through practical implementations.
- **Joining** the learning communities, e-interview the team behind and participate in HackFS hackathon to dig deeper into the technology stack (Decentralized storage network Part).
- **Conducting** a survey with purposing sampling, senior to managerial role-based Engineers with hands-on experience in on-premises and cloud to support the above methods.

The order of this project follows the same chronology of these models, the world has started with on-premises, then cloud and now we are witnessing the birth of the new model; Decentralized storage network.In each chapter, we have followed the same line in introducing some technologies that contributed to bringing this technology to existence to give the reader a general overview of the whole picture. The contribution of this project includes the following aspects.

- **Introducing** the paper and illustrating the outline.
- **Exploring** storage on-premises, articulating its state of the arts, and exploring the technical capabilities of XTREMIO,Isilon and ECS as storage samples.
- **Exploring** storage on Cloud, articulating its state of the arts, and exploring the technical capabilities of Azure storage and its tool set as an example.
- **Exploring** decentralized storage networks, articulating its state of the arts, and exploring the technical capabilities of IPFS and Filecoin as examples.
- **The result** and conclusion chapter which contains our outputs; pros and cons, technical comparison, survey result, recommendation, and final conclusion.
- **Three appendixces** that contains the practical implementations as follow:
  - **Appendix A:** ECS storage lab
  - **Appendix B:** Azure storage lab
  - **Appendix C:** IPFS lab

2 Storage on Premises

The term ”On Premises” refers to local hardware, which means that the data is stored on local servers, computers or other devices that are located within the physical confines of an enterprise , usually in the company’s data center . For example, a company may purchase a server on which to store data. After buying the server, the company sets it up at their headquarters and uploads their data. Because the server is locally operated, it’s considered on-premises data storage.

By installing and running software on hardware located within the premises of the company, internet technology (IT) staff has physical access to the data and can directly control the configuration, management and security of the computing infrastructure and data.

For the organizations whose core business is not information technology, the maintenance of an IT infrastructure is pretty expensive. Recruiting suitably skilled staff and arranging their trainings and certifications, software licenses and hardware infrastructure need huge investment and continuous maintenance and management responsibilities.Some organizations outsource the IT infrastructure in order to reduce the costs of its IT infrastructure and focus on the core business needs.

Another aspect of the cost involved in setting up and running a private cloud is the total cost of ownership ”TCO”, which not only involves the initial set up cost but also electricity cost, the cost of renewing the hardware, and staffing costs. [6]

2.1 What is a Data Center ?

A data center is a facility that centralizes an organization’s shared IT operations and equipment for the purposes of storing, processing, and disseminating data and applications. Because they house an organization’s most critical and proprietary assets, data centers are vital to the continuity of daily operations. Consequently, the security and reliability of data centers and their information are among any organization’s top priorities. [7]

Figure 2.1: Data Center
[3]

A data center typically consists of the following: [7]

- **Facility:** the usable space available for IT equipment. Providing round-the-clock access to information makes data centers some of the world's most energy-consuming facilities. Design to optimize space and environmental control to keep equipment within specific temperature/humidity ranges are both emphasized.
- **Support infrastructure:** equipment contributing to securely sustaining the highest availability possible. The Up time Institute has defined four tiers of data centers, with availability ranging from 99.671 Percent to 99.995 Percent. Some components for supporting infrastructure include:
  - Uninterruptible Power Sources (UPS) – battery banks, generators and redundant power sources.
  - Environmental control– computer room air conditioners (CRAC); heating, ventilation and air conditioning (HVAC) systems; and exhaust systems. Physical security systems – biometrics and video surveillance systems.
- **Core components:**equipment and software for IT operations and storage of data and applications. These may include storage systems; servers; network infrastructure, such as switches and routers; and various information security elements, such as firewalls.
- **Operations staff:** personnel available to monitor operations and maintain IT and infrastructure equipment around the clock.

Data centers are built to fulfill the below key characteristics as shown in the figure . Although those characteristics are applicable to almost all data center components. However, the details here mainly focus on storage systems.

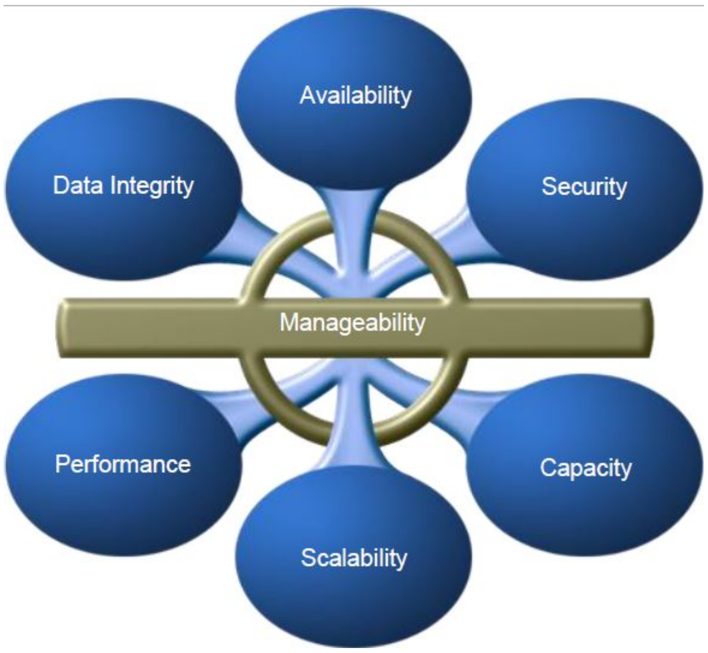


Figure 2.2: Data Center Characteristics
[3]

Data center characteristics are: [3]

- **Availability:** Availability of information as and when required should be ensured. Unavailability of information can severely affect business operations, lead to substantial financial losses, and damage the reputation of an organization.
- **Security:** Policies and procedures should be established, and control measures should be implemented to prevent unauthorized access to and alteration of information.
- **Capacity:** Data center operations require adequate resources to efficiently store and process large and increasing amounts of data. When capacity requirements increase, additional capacity should be provided either without interrupting the availability or with minimal disruption. Capacity may be managed by adding new resources or by reallocating existing resources.
- **Scalability:** Organizations may need to deploy additional resources such as compute systems, new applications, and databases to meet the growing requirements. Data center resources should scale to meet the changing requirements, without interrupting business operations.
- **Performance:** Data center components should provide optimal performance based on the required service levels.
- **Data integrity:** Data integrity refers to mechanisms, such as error correction codes or parity bits, which ensure that data is stored and retrieved exactly as it was received.
- **Manageability:** A data center should provide easy, flexible, and integrated management of all its components. Efficient manageability can be achieved through automation for reducing manual intervention in common, repeatable tasks.

2.2 Evolution of storage architecture

When it comes to business data storage systems, there are two major architectures:

- **Server-centric Architecture.**
- **Information-centric Architecture.**

Deciding which is right for your business depends on the following factors:

1. **Capacity:** How much data do you need to store?
2. **Scalability:** How much data will you need to store 5 to 10 years from now?
3. **Reliability:** Can your business survive without its data, files and applications? What would downtime do to your business?
4. **Backup and Recovery:** Where will you back up files and how often? What would happen if you lost files?

5. **Performance:** How many employees need to share/access or collaborate on files, from where (remote or in-house) and how often?

6. **Budget:** How much do you have to spend? IT Staff and Resources: Do you have a dedicated IT staff person to manage your system?

2.2.1 Server-centric Architecture

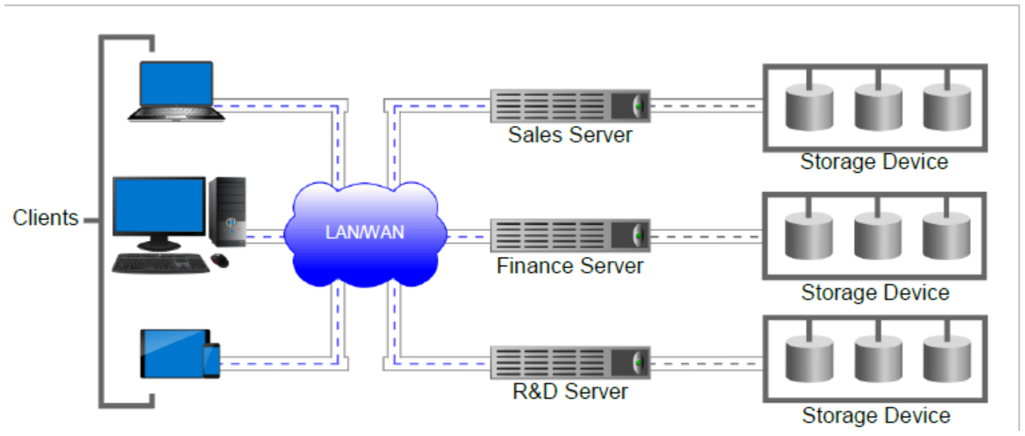


Figure 2.3: Server Centric Architecture [3]

Figure 2.3 shows an example of server-centric architecture: In this storage architecture,the digital storage system is directly attached to a server or workstation, without a network in between. A typical DAS ”Direct -attached Storage” system is made of a data storage device (for example enclosures holding a number of hard disk drives) connected directly to a computer through a host bus adapter, which in the past used to be SCSI, but these days more often eSATA, SAS or Fiber Channel. The most important differentiation between DAS and NAS is that between the computer and DAS there is no network device (like a hub, switch, or router).

- The servers of different departments in an organization are connected to the shared storage over a SAN.
- The clients connect to the servers over a LAN or a WAN.
- When a new server is deployed in the environment, storage is assigned to the server from the same shared pool of storage devices.
- The storage capacity can be increased dynamically and without impacting information availability by adding storage devices to the pool.
- This architecture improves the overall storage capacity utilization, while making management of information and storage more flexible and cost-effective.

The best Use Case Scenario is for small businesses that only need to share data locally, have a defined, non-growth budget to work with and have little to no IT support to maintain a complex system.While, the worst Use Case is for businesses that are growing quickly, need to scale quickly, need to share across distance and collaborate or support a lot of system users and activity at once.

2.2.2 Information-Centric Architecture

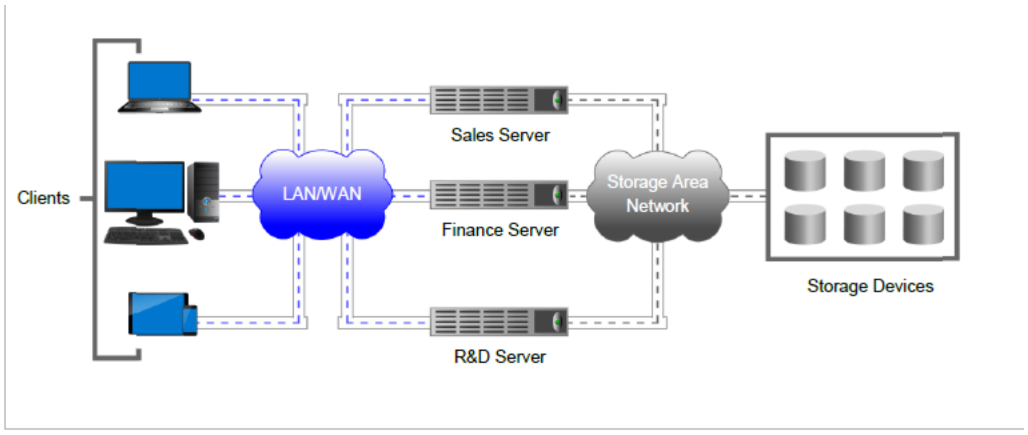


Figure 2.4: Information Centric Architecture
[3]

To overcome the challenges of the server-centric architecture, storage evolved to the information-centric architecture.In this storage architecture, storage devices exist independently of servers, and are managed centrally and shared between multiple compute systems.

Figure 2.4 above shows an example of information-centric architecture .

- The servers of different departments in an organization are connected to the shared storage over a SAN.
- The clients connect to the servers over a LAN or a WAN.
- When a new server is deployed in the environment, storage is assigned to the server from the same shared pool of storage devices.
- The storage capacity can be increased dynamically and without impacting information availability by adding storage devices to the pool. [8]

This architecture improves the overall storage capacity utilization, while making management of information and storage more flexible and cost-effective.

2.3 Software Defined Data Center

Software defined data center is an architectural approach to IT infrastructure that extends virtualization concepts such as abstraction, pooling, and automation to all of the data center's resources and services to achieve IT as a service. [3]

In an SDDC environment ,the compute, storage, networking, security, and availability services are pooled, aggregated, and delivered as a service. SDDC services are managed by intelligent, policy-driven software. [3]

SDDC is viewed as an important part in the progress towards a complete virtualized data center (VDC), and is the necessary foundational infrastructure for the modern data center.There can be software-defined compute (compute virtualization), software-defined network (SDN), and software-defined storage (SDS).

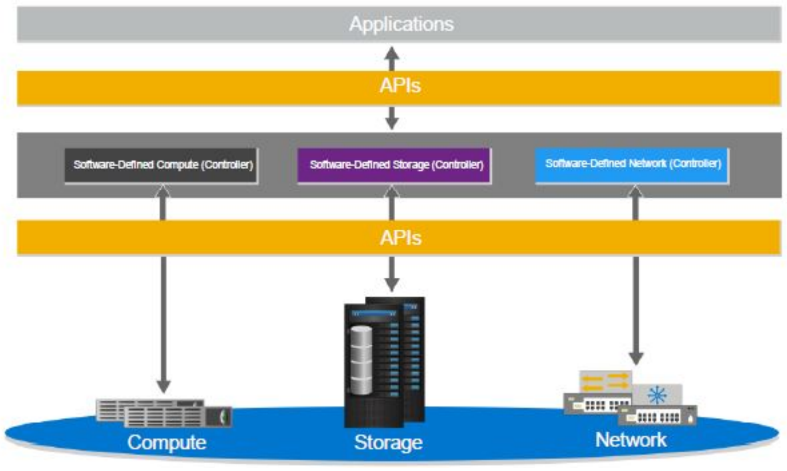

Applications
APIs
Software-Defined Compute (Controller)
Software-Defined Storage (Controller)
Software-Defined Network (Controller)
APIs
Compute
Storage
Network

Figure 2.5: Software Defined Data center Architecture
[3]

The software-defined approach decouples the control or management functions from the underlying hardware components and provides it to external software, where the external software takes over the control operations and enables the management of multi-vendor infrastructure components centrally. [9]

Any physical infrastructure component (compute, network, and storage) has a control path and a data path. The control path sets and manages the policies for the underlying resources, while the data path is responsible for the transmission of data.In the software-defined approach, it the control path and the data path are decoupled. By this way , the control path, resource management function operates at the control layer. This layer gives the ability to partition the resource pools, and manage them uniquely by policy. This decoupling of the control path and data path enables the centralization of data provisioning and management tasks through software that is external to the infrastructure components. The software runs on a centralized compute system or a stand-alone device, called the software-defined controller. The figure above shows the software-defined architecture, where the management function is separated from the underlying infrastructure components using controller. [9]

2.4 Modern Data Center Infrastructure

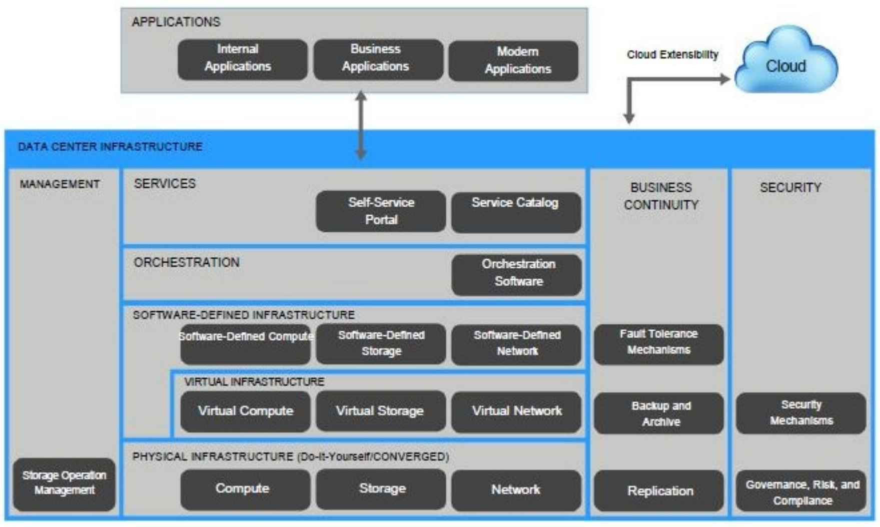


Figure 2.6: Modern Data Center Infrastructure
[3]

The figure 2.6 is a block diagram showing the core IT infrastructure components that builds up a modern data center.

Modern Data Center Infrastructure consists of five logical layers and three cross-layer functions. The five layers are:

- Physical infrastructure.
- Virtual infrastructure.
- Software-defined infrastructure.
- Orchestration.
- Services.

The three cross-layer functions are:

- Business continuity.
- Security.
- Management.

Business continuity and security functions include mechanisms and processes that are required to ensure a reliable and secure access to applications, information, and services. The management function allows efficient administration of the data center and the services for meeting business requirements.

- **Physical Infrastructure**:The physical infrastructure is the building block of the data center. It is the hardware used such as compute systems, storage systems, and networking devices. These equipment along with the operating systems, system software, protocols, and tools enable the physical equipment to perform their functions. The main function of physical infrastructure is to execute the requests generated by the virtual and software-defined infrastructure. In addition to storing data on the storage devices, performing compute-to-compute communication and creating backup copies of data.

- **Virtual Infrastructure**:The Virtual infrastructure abstracts the physical resources in the layer below and creates virtual resources for ”compute, storage and network”.This is achieved by using virtualization software that is deployed on compute systems, storage systems, and network devices. For example, storage virtualization software pools the capacity of multiple storage devices to create a single large storage capacity. Similarly, compute virtualization software pools the processing power and memory capacity of a physical compute system. This physical computes create an aggregation of the power of all processors (in megahertz) and all memory (in megabytes). Examples of virtual resources include virtual compute (virtual machines), virtual storage (LUNs), and virtual networks.

- **Software-Defined Infrastructure**: SDI refers to the operation and control of IT infrastructure entirely using software technologies and without involvement of the human element. Processes including infrastructure control, management, provisioning, configuration and other architectural operations are performed automatically via software as per application requirements and the defined operational policies. Since the changes are not dependent or limited to the human involvement, SDI enables intelligent infrastructure processes based on the changing IT operation requirements in real-time. The IT infrastructure therefore becomes intelligent, taking smart decisions on its own in order to meet the defined goals on SLAs, performance, security and other considerations. SDI allows the infrastructure to operate as a self-aware, self-healing, self-scaling and self-optimizing IT environment to enable truly agile business processes. [9]

- **Orchestration**:Data center orchestration software uses the automation of tasks to implement processes, such as deploying new servers. Automation solutions which orchestrate data center operations enable an agile DevOps approach for continual improvements to applications running in the data center. [10]

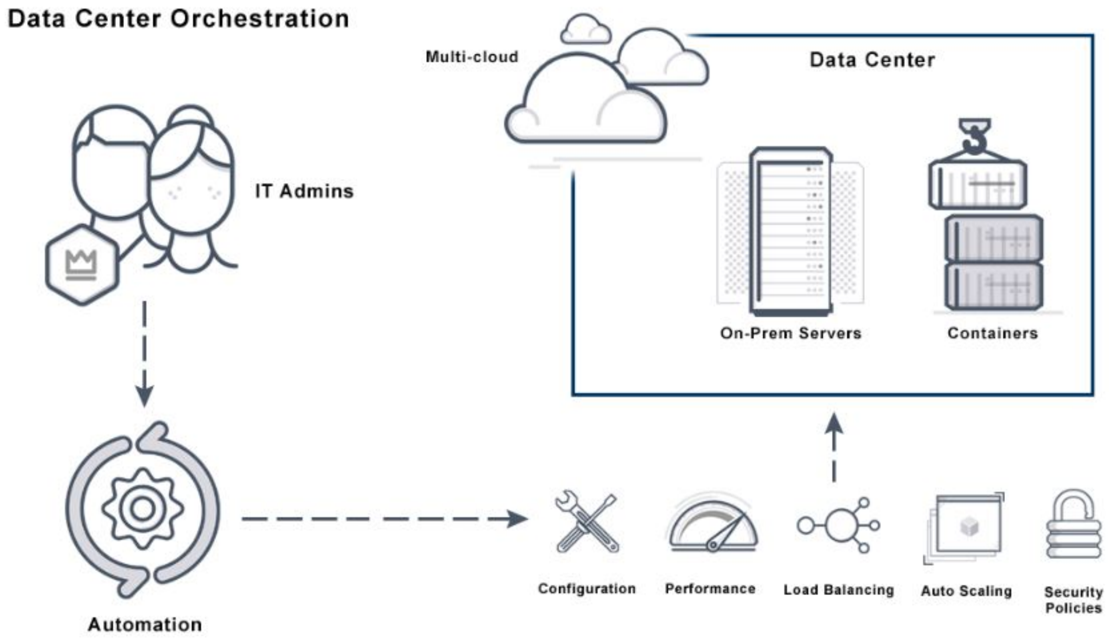


Figure 2.7: Data Center Orchestration
[10]

- **Services**:An IT service is a means of delivering IT resources to the end users to enable them to achieve the desired business results and outcomes without having any liabilities such as risks and costs associated with owning the resources. Examples of services are application hosting, storage capacity, file services, and email. The service layer is accessible to applications and end users. This layer includes a service catalog that presents the information about all the IT resources being offered as services. The service catalog is a database of information about the services and includes various information about the services, including the description of the services, the types of services, cost, supported SLAs, and security mechanisms. [3]

- **Business Continuity**:is the advance planning and preparation undertaken to ensure that an organization will have the capability to operate its critical business functions during emergency events. Events can include natural disasters, a business crisis, pandemic, workplace violence, or any event that results in a disruption of your business operation. It is important to remember that you should plan and prepare not only for events that will stop functions completely but for those that also have the potential to adversely impact services or functions. [11]

- **Security**:The security cross layer function supports all the infrastructure layers :physical, virtual, software-defined, orchestration, and service—to provide secure services to the consumers. Security specifies the adoption of administrative and technical mechanisms that mitigate or minimize the security threats and provide a secure data center environment.Examples of technical mechanisms include firewall, intrusion detection and prevention systems, and anti-virus software.

- **Management**:The management cross-layer function enables IT administrators to manage the data center infrastructure and services.Some of these tasks include handling of infrastructure configuration, resource provisioning, problem resolution, capacity, availability, and compliance conformance. This function supports all the layers to perform monitoring, management, and reporting for the entities of the infrastructure.

2.5 Intelligent Storage Systems

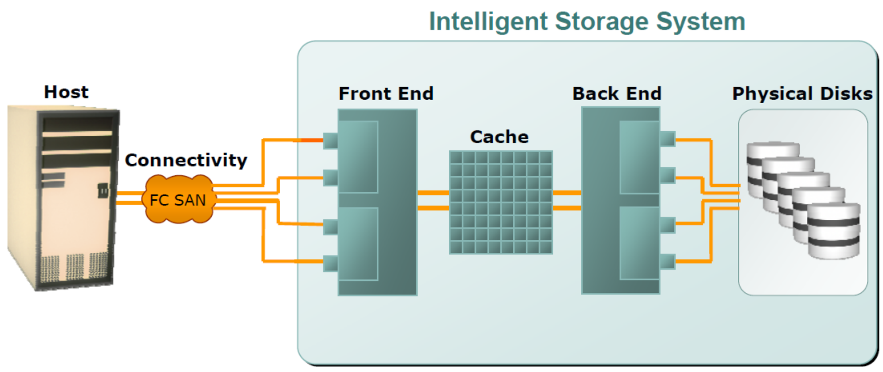


Figure 2.8: Intelligent Storage Systems
[12]

The intelligent storage system is a feature-rich RAID arrays that provide highly optimized I/O processing capabilities [13].These storage systems are configured with a huge amount of memory named cache which use sophisticated algorithms to meet the I/O requirements of performance-sensitive applications.

These applications require high levels of performance, availability, security, and scalability. Therefore, to meet the requirements of the applications, many vendors of intelligent storage systems now support SSDs , which are solid state drives that contain non-volatile flash memory hybrid drives.These drives are well suited for low-latency applications that require consistent, low (less than 1 millisecond) read/write response times. Encryption, compression, deduplication, and scale-out architecture.

Figure 2.9 shows the components of an intelligent storage system :

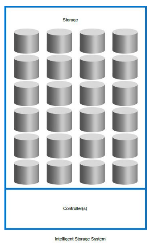
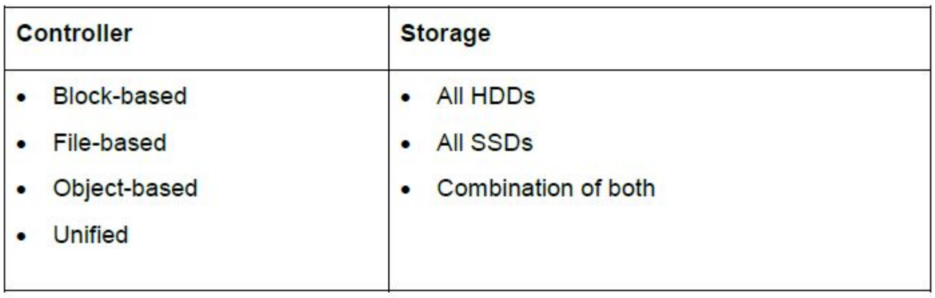


Figure 2.9: Intelligent Storage Systems
[3]

There are two main components of an intelligent storage system:Controller and Storage .

Figure 2.10: Intelligent Storage Systems



Controller	Storage
• Block-based • File-based • Object-based • Unified	• All HDDs • All SSDs • Combination of both

[3]

A storage system can have all hard disk drives, all solid state drives, or a combination of both. .The controller is a compute system that runs a purpose-built operating system that is responsible for performing several key functions for the storage system. Examples of such functions are:

- **Serving I/Os from the application servers**.
- **Storage Management**.
- **RAID protection ”Redundant Array of In dependant Disks”**: is a data storage structure enabling a data center to combine two or more physical storage devices, such as HDDs, SSDs, or both, into a logical unit that the attached system views as a single drive.RAID can either be hardware-based or software-based.
- **Local and remote replication:**Data replication is the process of storing the same data in multiple locations to improve data availability and accessibility.One common use of data replication is for disaster recovery, to ensure that an accurate backup exists at all times in case of a catastrophe, hardware failure, or a system breach where data is compromised.
- **Provisioning storage:**Storage provisioning is the process of assigning storage capacity to servers, computers, virtual machines or any other computing device.
- **Automated tiering:**Automated tiering is a quality of storage service feature that enables logical volumes or LUNs to span different tiers of storage and transparently move portions of the logical volume between tiers of storage to minimize storage costs and deliver consistent performance and throughput. [14]
- **Data compression:**Data compression is the process of encoding, restructuring or otherwise modifying data in order to reduce its size. Fundamentally, it involves re-encoding information using fewer bits than the original representation. [15]
- **Data encryption:**Data encryption means encrypting data while it passes to storage devices, such as individual hard disks, tape drives, or the libraries and arrays that contain them. Using storage level encryption along with database and file encryption goes a long way toward offsetting the risk of losing your data.

- **Intelligent cache management:**Cache is a type of memory that is used to increase the speed of data access. Normally, the data required for any process resides in the main memory. However, it is transferred to the cache memory temporarily if it is used frequently enough.

An intelligent storage system typically has more than one controller for redundancy. Each controller consists of one or more processors and a certain amount of cache memory to process a large number of I/O requests. These controllers are connected to the compute system either directly or via a storage network. The controllers receive I/O requests from the compute systems that are read or written from/to the storage by the controller. Depending on the type of the data access method used for a storage system, the controller can either be classified as block-based, file-based, object-based, or unified.

2.6 RAID Techniques

RAID ”Redundant Arrays of Independent Disks ” is a technique that combines multiple disk drives into a logical unit (RAID set) in order to achieve data protection, high performance, or both.

RAID is typically implemented on the compute system or on the storage system. The key functions of a RAID controller are: management and control of drive aggregations, translation of I/O requests between logical and physical drives, and data regeneration in case of drive failures.

Software RAID uses compute system-based software to provide RAID functions and is implemented at the operating-system level. Software RAID implementations offer cost and simplicity benefits when compared with hardware RAID. However, they have the following limitations [3]:

- **Performance:** Software RAID affects the overall system performance. This is due to additional CPU cycles required to perform RAID calculations.
- **Supported features:** Software RAID does not support all RAID levels.
- **Operating system compatibility:** Software RAID is tied to the operating system; hence, upgrades to software RAID or to the operating system should be validated for compatibility. This leads to inflexibility in the data-processing environment.

There are seven RAID level groups :

RAID 0 implements striping, which means that the data is divided into strips and each strip is written on different hard disks in the disk array. RAID 0 requires a minimum of two disk drives to be implemented. The design is simple and easy to implement. However, since there is no redundancy, it doesn’t provide fault tolerance. If even one drive fails, data across the drive array will be lost.

RAID 1 implements mirroring , which means that data is duplicated in both the disks. On disk failure, the recovery is easier as compared to other RAID levels. Data just needs to be copied into the new disk. Write performance is worse than writing on

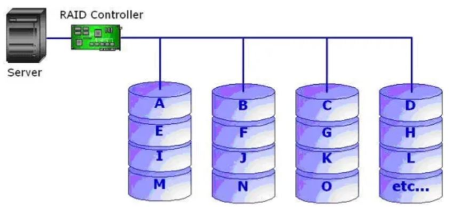


Figure 2.11: RAID 0 - Striped Disk Array without Fault Tolerance
[16]

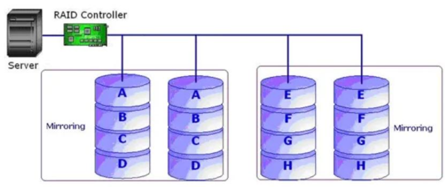


Figure 2.12: RAID 1-Mirroring
[16]

a single disk as multiple writes have to be done, The design is simple. However, the major drawback of RAID 1 is that actual disk utilization is only 50 percent as half of the disks are utilized for mirroring.

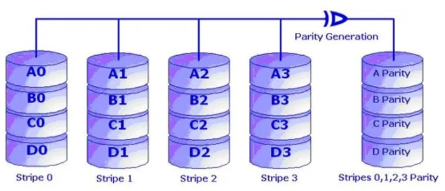


Figure 2.13: RAID 3-Striping with Bit Level Parity
[16]

RAID 3 implements byte level striping with parity. It requires a minimum of 3 disks to be implemented. Data to be written is divided into stripes and stripe parity is calculated for every write operation. The stripe parity is stored on a separate parity disk.It provides fault tolerance and disk usage is better than that of mirroring. Controller design is quite complex. Write operations are slow as there are overheads of parity calculation and writing parity to a separate disk. Read operations are faster as compared to write.

RAID 4 implements block level striping with parity. It requires a minimum of 3 disks to be implemented. It uses block-level data striping and a dedicated disk for storing parity bits. It does not require synchronized spinning, and each disk functions independently when single data blocks are requested. This is in contrast to RAID 3, which stripes at block-level, versus bit-level.This configuration requires at least three disks.

RAID 5 is the most useful RAID mode when a larger number of physical disks are combined, and redundant information in terms of parity is still maintained. To

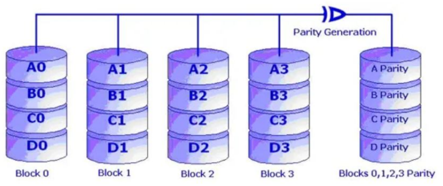


Figure 2.14: RAID 4-Striping with Block Level Parity
[16]

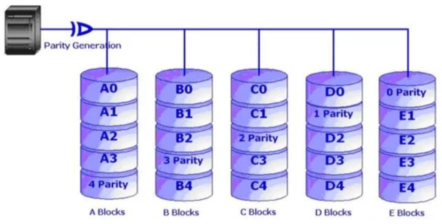


Figure 2.15: RAID 5-Distributed Parity
[16]

implement RAID-5 a minimum of 3 disks are required. The difference between RAID-5 and RAID-4 is that the parity information is distributed evenly in the drives, thus avoiding the bottleneck problem in RAID-4. RAID-5 also can sustain maximum one disk failure at a time. Reading is similar to RAID 0 but writes can be expensive. Writing to disk array requires reading of whole row, calculating parity and then rewriting data which makes the writing a bit slow. The drawbacks of RAID-5 are that the controller design is very complex and rebuilding data in case of disk failure is difficult if compared to RAID level 1.

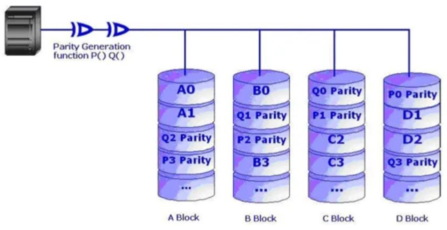


Figure 2.16: RAID 6-Distributed Dual Parity

RAID-6 is an extension of RAID-5 to provide additional fault tolerance by using dual distributed parity schemes. Dual parity scheme helps survive two disk failures at a time without data loss.RAID-6 is a perfect solution for mission critical applications. Read performances are same as that in RAID-5 but writes are slow due to dual parity calculation overhead. Disk usage is less as compared to RAID-5 as more space is required to store two parities.

2.7 Block Based Storage Systems

Block Level Storage refers specifically to saving data in volumes called blocks. It is when a raw volume of data storage is presented to a server, usually from a storage attached network (SAN), and each volume block can function as an individual hard drive or storage repository.

Block storage can be presented to any operating system as a mounted drive volume. A storage consultant can allocate almost any size of data volume to a server. The end user will simply see this as a mapped drive or a local hard drive on the server's operating system. The server operating systems can access the storage blocks over high-speed Fiber Channel or iSCSI connectivity options.

Figure 2.17 shows an example of a block storage array product is XtremIO :

Figure 2.17: XtremIO
[17]

XtremIO is one of Dell EMC's high end all block flash storage arrays that is designed for high performance. Supplying consistent, predictable sub-millisecond latency with high IOPs is the foundation of XtremIO X2. XtremIO X2 is scalable where performance, memory, and capacity increase linearly. XtremIO X2 offers advanced features and a rich set of data services. Inline data reduction where it efficiently stores only unique 16 kB blocks of data. This provides savings on storage cost by avoiding writes of duplicate blocks to Flash. Since XtremIO X2 is a fully thin provisioned storage system, even more storage savings are realized. The XtremIO X2 is simple and straight forward to use. The array has an easy to use graphical interface, simple volume provisioning, and requires minimal design and implementation efforts.

What makes XtremIOX2 unique?

1. XtremIOX2s unique content-based metadata engine coupled with its scale-out architecture delivers the unique value. You grow an XtremIOX2 cluster by non-disruptively adding more X-Bricks, linearly increasing both capacity and performance at the same time.

2. The volumes are always thin-provisioned, the only writes performed to the disks are for data that are globally unique across the entire cluster.

3. All the data services are inline, all the time. So as the writes come in, the metadata engine looks at the content of the data blocks. It performs the writes to the SSDs only if the cluster did not see the data before. For duplicate data, it just acknowledges the writes to the host with appropriate updates to in-memory metadata without actually performing any writes to the SSDs. This inline deduplication saves tremendous capacity for persistent desktops without affecting performance at all.

4. Inline compression adds to XtremIOX2s data reduction efficiencies.

5. XtremIOX2 has a proprietary Flash-based data protection algorithm (XDP) that offers better than RAID-10 performance with RAID 5 capacity savings.

6. XtremIOX2s differentiated agile copy data service helps you provision and deploy desktops at a much faster rate than any other Flash-based solution on the market. [18]

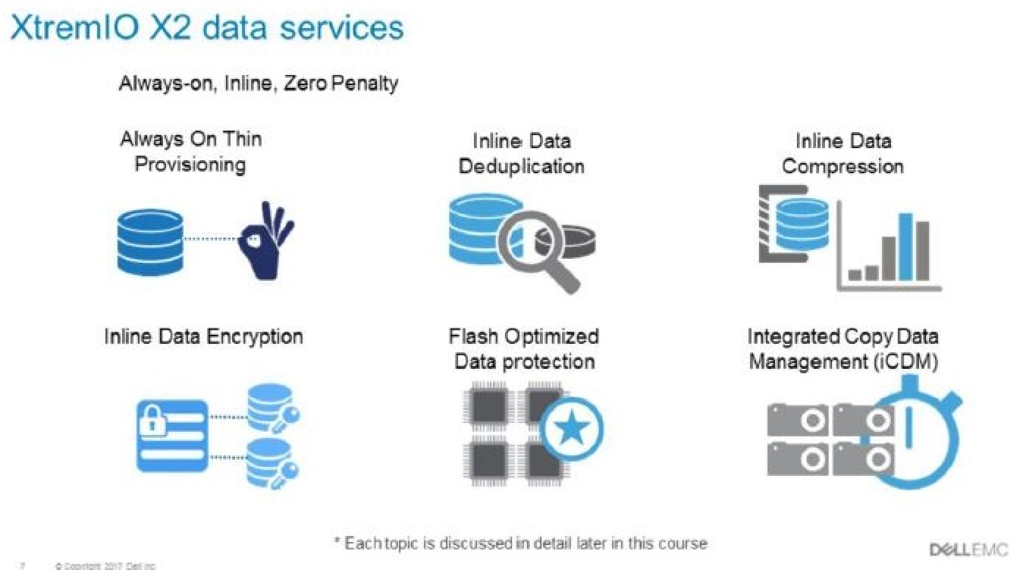

XtremIO X2 data services
Always-on, Inline, Zero Penalty
Always On Thin Provisioning
Inline Data Deduplication
Inline Data Compression
Inline Data Encryption
Flash Optimized Data protection
Integrated Copy Data Management (iCDM)
* Each topic is discussed in detail later in this course
DELLEMC

Figure 2.18: XtremIO
[17]

2.8 File based Storage System

File-based storage is a progressive storage technique used to sort out and store data especially unstructured data on a PC's hard drive or on network-attached storage (NAS) device. As shown in the below figure , data is stored in files, the files are organized in folders, and the folders are organized under a hierarchy of directories and sub-directories. To locate a file, all you or your computer system need is the path—from directory to sub-directory to folder to file. Data can be accessed using the Network File System (NFS) protocol for Unix or Linux, or the Server Message Block (SMB) protocol for Microsoft Windows.

An example of a File-based storage array product is Isilon:

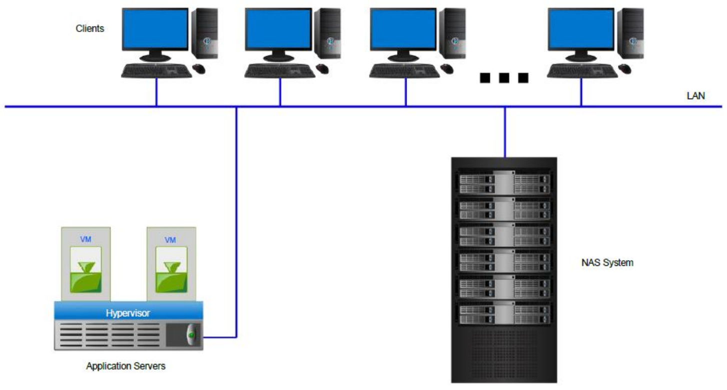


Figure 2.19: File based storage
[3]

Dell EMC Isilon all-flash storage platforms, powered by the OneFS operating system, provide a powerful yet simple scale-out storage architecture to speed access to massive amounts of unstructured data, while dramatically reducing cost and complexity. With a highly dense design that contains 4 nodes within a single 4U chassis, all-flash platforms deliver extreme performance and efficiency for your most demanding unstructured data applications and workloads. The all-flash platforms are available in 2 product lines:

- **Isilon F800:** Provides massive performance and capacity. It delivers up to 250,000 IOPS and 15 GB/s aggregate throughput in a single chassis configuration and up to 15.75M IOPS and 945 GB/s of aggregate throughput in a 252node5 cluster. Each chassis houses 60 SSDs with a capacity choice of 1.6 TB, 3.2 TB, 3.84 TB, 7.68 TB or 15.36 TB per drive. This allows you to scale raw storage capacity1 from 96 TB to 924 TB in a single 4U chassis and up to 58 PB5 in a single cluster.

- **Isilon F810:** Provides massive performance and capacity along with inline data compression and deduplication capabilities to deliver extreme efficiency. The F810 delivers up to 250,000 IOPS and 15 GB/sec aggregate throughput in a single chassis configuration and up to 15.75M IOPS and 945 GB/s of aggregate throughput in a 252 node cluster2 . Each F810 chassis houses 60 SSDs with a capacity choice of 3.84 TB, 7.68 TB or 15.36 TB per drive. This allows you to scale raw storage capacity from 230 TB to 924 TB in a 4U chassis and up to 58 PB of raw storage in a single cluster. Depending on your specific data set and workload, F810 inline data compression and deduplication delivers up to a 3:1 reduction in storage requirements, this increasing the effective capacity up to 138 PB per cluster.

  1. **Efficiency:** Isilon scale-out storage delivers up to 80 percent storage utilization versus about 50 percent for traditional NAS platforms. SmartDedupe data deduplication software enhances storage efficiency to reduce your physical storage requirements by up to 35 percent. A policy-based, automated tiering option allows you to optimize storage resources and further lower costs. In addition to these advantages that apply to all Isilon platforms, F810 all-flash offers inline data compression and data dedupli-

cation to further reduce data storage infrastructure requirements, increase density and lower costs.

2. **Flexibility:** Powered by the OneFS operating system, Isilon all-flash storage supports all major protocols and data access methods including NFS, SMB, HDFS, HTTP, and FTP. This means that you can support a wide range of unstructured data applications and workloads on a single storage platform.
3. **Data protection:** All-flash platforms are highly resilient and offers N+1 through N+4 redundancy. With OneFS you may also choose from a variety of efficient and proven enterprise data backup and disaster recovery options.
4. **Security:** All-flash platforms offer a broad range of robust security options including FIPS 140-2 level 2 self-encrypting drives, rolebased access control (RBAC), secure access zones, SEC 17a-4 compliant WORM data immutability, and file system auditing support. [19]

Figure 2.20: File based storage

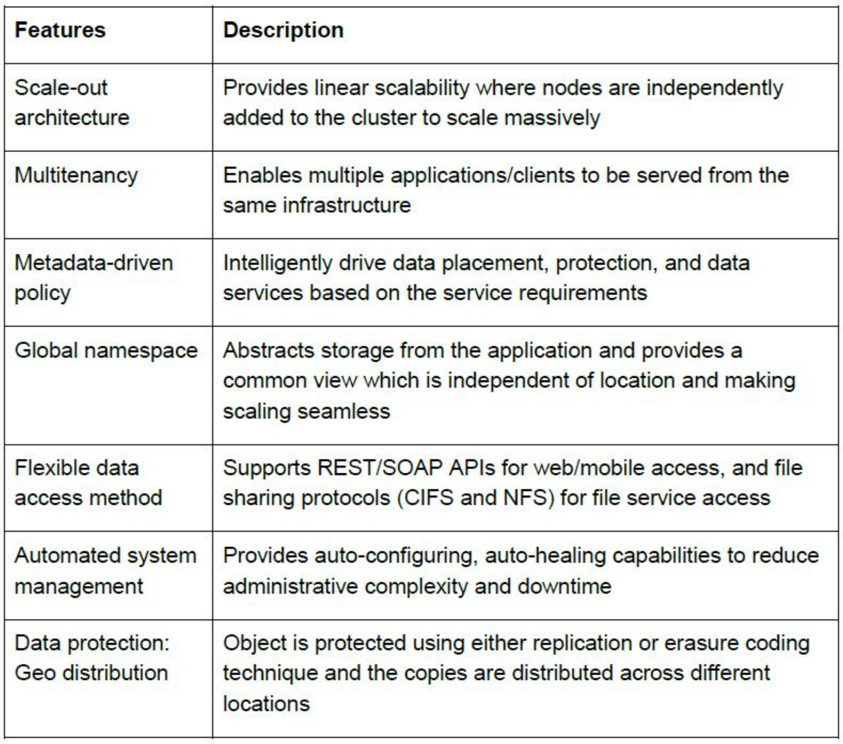
2.9 Object Based Storage Storage Systems

Object based storage is an interface for data storage in order to manage huge volumes of unstructured data. This is data that does not correspond to a standard relational database of rows and columns or cannot be easily structured into that. This includes email, videos, photos, web pages, audio files, sensor data, and other types of media and web content (textual or non-textual). This content streams continuously from social media, search engines, mobile, and "smart" devices. Enterprises are finding it challenging to efficiently store and manage this unprecedented volume of data. Object based storage has emerged as the preferred method for data archiving and backup. It offers a level of scalability not possible with traditional file- or block-based storage. With object based storage, one can store and manage data volumes on the order of terabytes (TBs), petabytes (PBs), and even larger.

Objects are discrete units of data that are stored in a structurally flat data environment. There are no folders, directories, or complex hierarchies as in a file-based system.

Each object is a simple, self-contained repository that includes the data, metadata (descriptive information associated with an object), and a unique identifying ID number, instead of a file name and file path. This information enables an application to locate and access the object. One can aggregate object storage devices into larger storage pools and distribute these storage pools across locations. Data in an object storage system are accessed via application programming interfaces (APIs). The native API for object storage is an HTTP-based RESTful API. These APIs query an object's metadata to locate the desired data via the Internet from anywhere, on any device. RESTful APIs use HTTP commands like "PUT" or "POST" to upload an object, "GET" to retrieve an object, and "DELETE" to remove it.

Figure 2.21: Object Based Storage Benefits



Features	Description
Scale-out architecture	Provides linear scalability where nodes are independently added to the cluster to scale massively
Multitenancy	Enables multiple applications/clients to be served from the same infrastructure
Metadata-driven policy	Intelligently drive data placement, protection, and data services based on the service requirements
Global namespace	Abstracts storage from the application and provides a common view which is independent of location and making scaling seamless
Flexible data access method	Supports REST/SOAP APIs for web/mobile access, and file sharing protocols (CIFS and NFS) for file service access
Automated system management	Provides auto-configuring, auto-healing capabilities to reduce administrative complexity and downtime
Data protection: Geo distribution	Object is protected using either replication or erasure coding technique and the copies are distributed across different locations

An example of a Object-based storage array product is ECS:

ECS provides significant value for enterprises and service providers seeking a platform architected to support rapid data growth. The main advantages and features of ECS that enable enterprises to globally manage and store distributed content at scale include: [20]

- **Cloud Scale** - ECS is an object storage platform for both traditional and next-gen workloads. ECS's software-defined layered architecture promotes limitless scalability. Feature highlights are:
  1. Globally distributed object infrastructure.
  2. Exabyte+ scale without limits on storage pool, cluster or federated environment capacity.
  3. No limits exist on the number of objects in a system or name space .

4. Efficient at both small and large file workloads with no limits to object size.

- **Flexible Deployment** - ECS has unmatched flexibility with features such as:
  1. Appliance deployment.
  2. Software-only deployment with support for certified or custom industry standard hardware.
  3. Multi-protocol support: Object (S3, Swift, Atmos, CAS) and File (HDFS, NFSv3).
  4. Multiple workloads: Modern apps and long-term archive.
  5. Secondary storage for Data Domain Cloud Tier and Isilon using Cloud-Pools.
  6. Non-disruptive upgrade paths from previous to current generation ECS models.
- **Enterprise Grade** - ECS provides customers more control of their data assets with enterprise class storage in a secure and compliant system with features such as:
  1. Data-at-rest (D@RE) with key rotation and external key management.
  2. Encrypted inter-site communication
  3. Disables ports 9101/9206 by default to empowers organizations to meet compliance policies.
  4. Reporting, policy- and event-based record retention and platform hardening for SEC Rule 17a4(f) compliance including advanced retention management such as litigation hold and min-max governance.
  5. Compliance with Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) hardening guidelines.
  6. Authentication, authorization and access controls with Active directory and LDAP.
  7. Integration with monitoring and alerting infrastructure (SNMP traps and SYSLOG).
  8. Enhanced enterprise capabilities (multi-tenancy, capacity monitoring and alerting).
- **TCO Reduction** - ECS can dramatically reduce Total Cost of Ownership (TCO) relative to both traditional storage and public cloud storage. It even offers a lower TCO than tape for long-term retention.

3 Storage On Cloud

According to NIST (National Institute of Standard and Technology) definition, ”cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” [21]

People may think that the concept of cloud computing appeared in the 21st century although it goes back over 60 years.Cloud computing has evolved gradually through a number of phases: [22]

In 1960,the computer scientist John McCarthy, shared a concept of time sharing, and enabled Organization to use expensive resources simultaneously.

In 1969,J.C.R. Licklider,who helped develop the ARPANET (Advanced Research Projects Agency Network), introduced the idea of an “Intergalactic Computer Network” or “Galactic Network” that aimed to let everybody on the earth to be able to interconnect and access data at any site from anywhere (a concept that is similar to the Internet nowadays).

In 1970,it became possible to run more than one Operating System simultaneously in an isolated environment which is the virtualization concept. It was possible to run a completely different Computer (virtual machine) inside a different Operating System.

In 1997,Prof. Ramnath Chellappa in Dallas introduced the first definition of the term “Cloud Computing” – “A computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits alone.”

In 1999, Salesforce.com an american company introduced the ability to deliver enterprise-level application on the Internet using a simple website.

In 2003, hypervisor, a software system that allows the execution of multiple virtual guest operating systems simultaneously on a single device,was first introduced.

In 2006, Amazon increased its cloud services. by introducing Elastic Compute cloud (EC2), which allowed users to access computers and run their own applications on them, all on the cloud. Then it launched Simple Storage Service (S3). This introduced the pay-as-you-go model to both users and the industry as a whole.

In 2007,MIT student created file hosting service that is called Dropbox which offers file storage and synchronization.

In 2009,Microsoft introduced its cloud computing platform,Windows Azure and the same year google introduced one of the famous browser based enterprise applications which is Google Apps .

The following figure shows the movement of the organization from traditional data center on-premises architecture to be on the cloud according to the Gartner estimation in 2019.

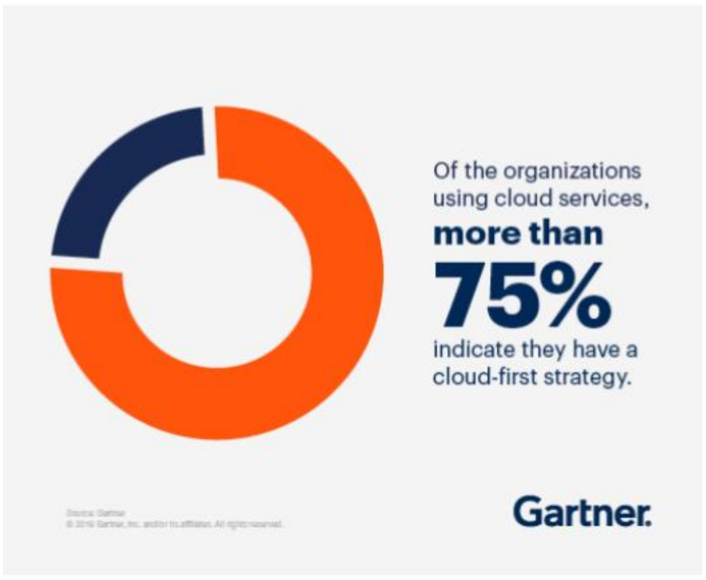


Figure 3.1: Gartner estimation for the growth of the cloud technology [23]

3.1 Cloud characteristics

NIST (National Institute of Standard and Technology) defines five essential characteristics for cloud computing:

- **On-demand self-service:** "A consumer can unilaterally provision computing capabilities, such as server time or networked storage, as needed automatically without requiring human interaction with each service provider. [24]
- **Broad network access:** "Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g.,mobile phones, tablets, laptops, and workstations). [24]
- **Resource pooling:** "The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence. In that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, and network bandwidth" [24].
- **Rapid elasticity:**"Capabilities can be rapidly and elastically provisioned, in some cases automatically, to scale rapidly outward and inward commensurate with de-

mand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time." [24]

- **Measured service:** "Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service." [24].Metering service also introduces the concept of **pay-per-use** or **Pay-as-you-Go** to the consumer with a better sense of resource consumption and provides transparency in billing that meet the service level agreement.

3.2 Cloud Service Models

Cloud computing offers three different service models with different capabilities and are suitable for different consumers.these service models are:

- **Infrastructure as a Service:** "The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (for example,host firewalls)." [24] One of the famous examples for IaaS is AWS(Amazon Web Services) EC2.

- **Platform as a Service:** "The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment." [24] .One of the famous examples for PaaS is Google App Engine.

- **Software as a Service:** "The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (for example, web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings." [24] .One of the famous examples of SaaS is Dropbox.

The below figure shows the customer management role in case of on-premises,IaaS,PaaS and SaaS.

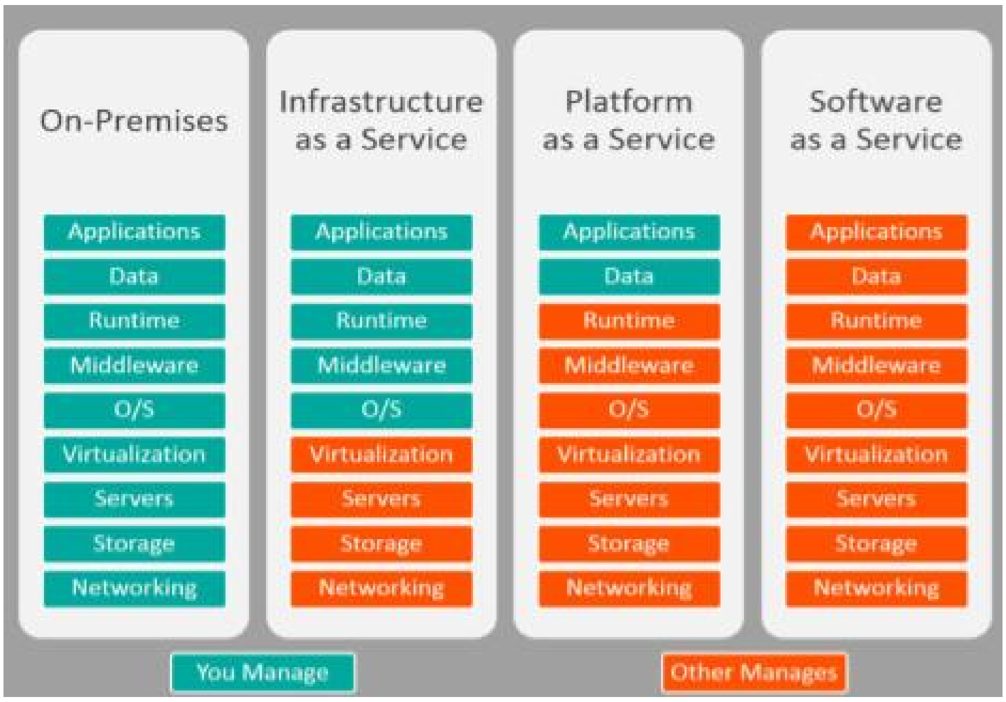


Figure 3.2: On-premise vs Cloud Computing service models [25]

3.3 Cloud Deployment Models

A cloud deployment model is a model that shows how cloud infrastructure is built, managed, and accessed. In SP 800-145, [24] defines four primary cloud deployment models for the cloud computing which are:

- **Public cloud:** “The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.” [24].
- **Private cloud:** “The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (for example, business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.” [24]. On the private cloud,the cloud services are dedicated to the departments and business units within this organization.
- **Community cloud:** “The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (for example, mission,security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some

combination of them, and it may exist on or off premises.” [24] A community cloud is a cloud infrastructure that is established to be used by a group of organizations that share common goals or requirements.

– **Hybrid cloud:** “The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound by standardized or proprietary technology that enables data and application portability (for example, cloud bursting for load balancing between clouds.)” – [24]

The following figure shows the difference between Public Cloud,Private Cloud, Community Cloud and Hybrid Cloud.

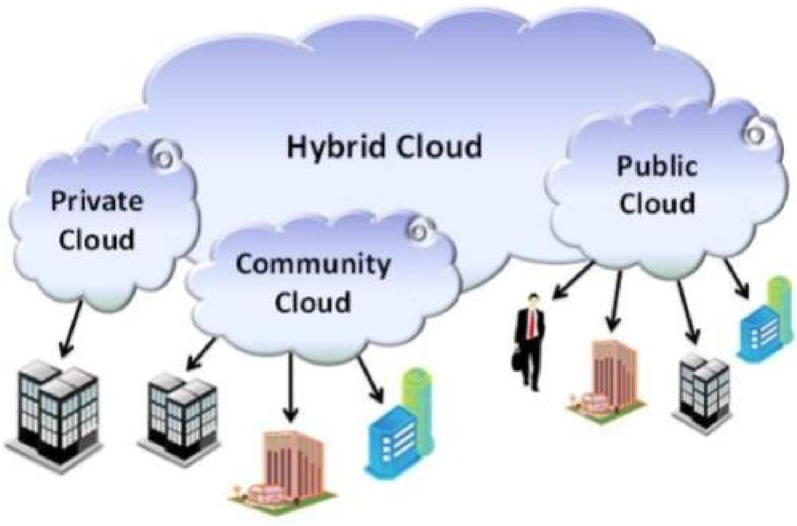


Figure 3.3: Cloud Computing Deployment model
[26]

3.4 Cloud computing advantages and disadvantages

Cloud computing offers your business numerous advantages. It permits you to set up a virtual office to give you the ability of associating with your business from anywhere and at any time. With the developing number of web-enabled devices such as(smartphones, tablets) cloud computing makes accessing your data much easier. There are many benefits that let any organization to move from traditional environment to the cloud such as:

- **Business agility:**In a traditional environment,the process of adding or having new IT resources may take hard procedures and approvals which leads the operations to be delayed and can increase time-to market.cloud computing,provides quick resource provisioning at any time that reduces the time to market and responds more quickly for the market changing.
- **Reduced IT costs**:In a traditional environment, resources are dedicated to specific business applications.In cloud computing provides consumers pay only for

the resources needed to be accessed in the cloud based on pay-per-use or pay-as-you go.which leads to decrease consumer's IT capital expenditure (CAPEX) as investment is done for the required resources.

- **High availability**:Cloud computing ensures resource availability at different levels by using redundancy techniques for compute systems, network paths, and storage equipment, along with clustered software which allows fault tolerance for cloud deployments. Redundancy techniques can handle multiple data centers located in different geographic regions, which prevents losing data due to regional failures.
- **Business continuity:**One of the important things that can affect the organization's reputations is loss of IT service availability which also leads to crucial financial losses to organizations .Cloud business continuity can reduce the effect of downtime using the disaster recovery concept which aims to have a remote secondary site.organization can use cloud-based backup to preserve additional copies of their data that can be retrieved in the case of an outage.
- **Flexible scaling:**In cloud computing it is easy to scale up IT resources to meet workload demand without the need of buying additional compute systems as consumers use shared pool of resources and once the consumer release the compute resources after the task is completed and thus other consumers can use this released resource using the concept of pay for what you use.
- **Flexibility of access:**In a traditional environment,dedicated devices such as desktop are used to access the IT resources .In cloud computing,we can access from any place the application using any device such as mobile,laptop,desktop and so on.

Although there are a lot of organization that move to cloud there are some disadvantage that face the organizations when moving to the cloud which are:

- **Security:** Although cloud providers try always to secure their infrastructure there is still an issue with the security. One of the latest examples of this was the breach that happened to google cloud database on march 2020, about 800 gigabytes of personal user information were breached that put a large number of people at risk since more than 200 million detailed user records. [27]
- **The need of high speed internet:**To have the ability to access the cloud you need a stable and high speed internet that gives you the ability to download or upload files on the cloud.
- **Vendor Lock-in:**Migration from one cloud service provider to another you may face the problem of interoperability and compatibility between different vendors.

3.5 Cloud Service providers

Cloud computing is distinguished by the capability of provisioning computing resources automatically and in a-fast way for organizations that don't own Data Centers which in turn removing the cost of installing and maintaining server operations

,therefore cloud computing was adopted by various companies Microsoft ,Amazon and Google that will be discussed later in this section.

3.5.1 Azure

Microsoft Azure is a platform of cloud computing services that contains technologies and solutions from Microsoft and other companies. Instead of establishing or leasing Data centers, Azure's billing is based on what resources do you use and consume giving you the flexibility to choose any service that it provides from a catalogue service. Pricing depends on various types of services, such as storage types, and region location from which your Azure instances running.

In October 2008 The Azure platform was declared, and became available in the market in February 2010. It was known for Windows Azure, after that it was renamed to Microsoft Azure in July 2014. Other services has been appended to regions since launching.

Azure attracts organizations that are running Microsoft solutions like active directory ,exchange and SharePoint especially if these companies have to upgrade their operating system from windows 2008 to 2019,they found that migrating to cloud would be easy and transparent take place with no issues as long as exchange and SharePoint are available as software as a service.

Microsoft's Azure for Students program gives a credit of 100 dollars available for 12 months, beside access to more than 25 free products, including Virtual Machines, File Storage, and SQL Databases for the first 12 months, while other products are always free.

Blob storage, file service, table, queue, Virtual machines Azure, Cosmos DB, and Azure Active Directory are all examples of services supported by Azure. [28]

3.5.2 AWS

AWS is a cloud computing services and applications offered by Amazon.com. The most popular of these services include Elastic Compute Cloud (EC2) and Simple Storage Service (S3) companies like "Netflix" have moved their business to AWS.

The AWS platform begins at 2006, with limited regions and services then new types of services was growing since launch.Presently, AWS services are available in distinct global "regions": US East (Ohio and Northern Virginia), US West (Oregon and Northern California), Canada, (Montreal), Brazil (São Paulo), England (London), EU (Ireland, France, Italy, Sweden, and Germany), Middle East (Bahrain), Africa (Cape Town), Asia Pacific (Singapore, Sydney, Tokyo, Seoul, Osaka, Mumbai, and Hong Kong), and Mainland China (Beijing, Ningxia).

AWS attracts any organization for using different services in AWS such as using S3 Glacier for offsite backups. While AWS begins as a cloud replacement for simple storage and compute operations, it has services that can meet any use case, with dedicated services for databases, IoT development, business productivity, messaging, game development, virtual desktops, analytics, machine learning.

Developers have the option to begin with AWS through the Free Tier, which is provided to anyone with no constraints or difficulties for the first 12 months. It features 750 hours per month of EC2 t.2 micro instances of Linux or Windows, as well as 5 GB of standard storage in S3 with 20,000 GET and 2,000 PUT requests.

Beside EC2 and S3 services, Relational Database Service (RDS) is a scalable database server that supports MySQL/MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server, as well as Amazon's own Aurora implementation of MySQL. Similarly, Dynamo DB offers scalable NoSQL database support. [29]

3.5.3 Google Cloud

Google App Engine was published in April 2008 as a Platform as a Service (PaaS) resource to help the developers to program and host applications on Google's infrastructure in order to reach the increasing trend in web applications. App Engine came on the play in September 2011, in 2013 the Google Cloud Platform name was announced.

Google Cloud Platform introduces a set of tools that Google cloud developed were intended for for internal use. This caused a big problem. Google designed cloud services for beginners and Small Business, gives an offer beginning with 100,000 dollars of Cloud Platform credits to startups. GCP was not flexible and up-to-date with diverse use cases.

As mentioned, Google's first foray into cloud services was the Google App Engine back in 2008. The initial appearance of Google platform was the Google App Engine in 2008 as discussed earlier. After 2 years, extending a storage layer has been declared by Google. In 2012, a partner program was started to generate the platform. BigQuery the compute Engine showed up, cloud SQL and the rest of the tools that form today's Google Cloud Platform.

Google provides users a free tier for Cloud Platform, beside a free 12-month trial with credit for organizations that want to study the cloud and know all the available features. free credits are given by Google to select startups working with big organizations through the Google Cloud for Startups promotion. Google gives a tool for live migrations (simply titled Live Migration), which let the virtual machine instance to continue to operate in case a host failure.

Regarding storage;Cloud Storage, Persistent Disk, Cloud File store are examples of services provided by Cloud. [30]

The following figure shows the position of cloud service provider according to Gartner chart for July 2019.

3.6 Storage on cloud

Cloud storage is a service model which provides storing and transmission of data on remote storage systems enabling the users to access it over a network(internet). Cloud Storage systems can be accessed from any location and can be accessed from any device .Data on these remote storage systems can be managed and backed up.The payment for storing data on cloud is based on the concept of pay-as-you go or pay-as-you consume.

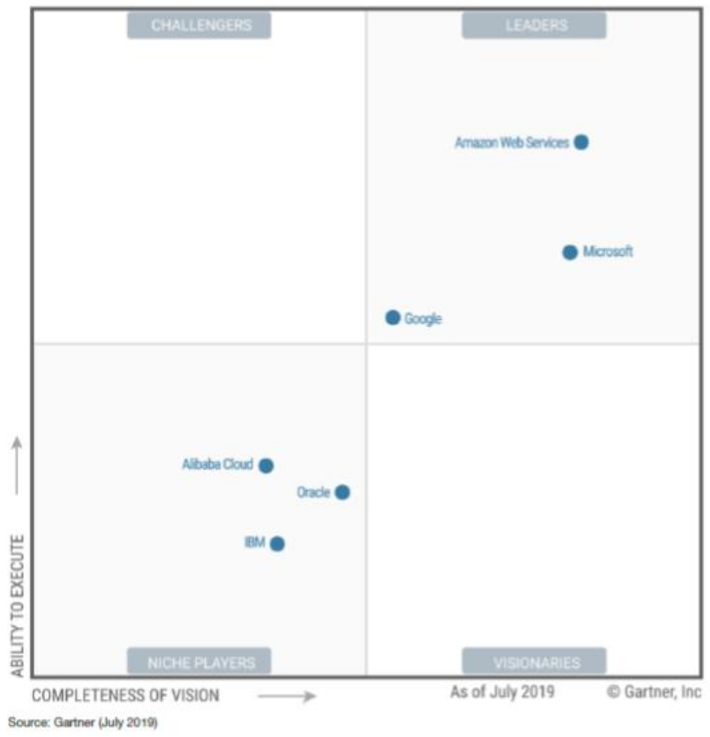


Figure 3.4: Gartner analysis for the growth of cloud service provider in the market [31]

There are three types of storage models: public cloud storage that is used with the unstructured data, private cloud storage that offers more control over the data by using a company firewall,hybrid cloud storage which is a combination of public and private cloud services that offers more flexibility.As mentioned before, cloud computing in general has its pros and cons also the cloud storage has the advantages and disadvantages.

Cloud storage benefits:

- **Pay as you use:**consumers pay only for what they use from the resources which reduce the cost as there is no need for the consumers to buy new storage devices or pay for the electricity of the powering system.
- **Data Redundancy:**cloud storage vendors enable high durability and availability for the data by having multiple copies of data on different data centers and also in different geographical regions.
- **Data Tiering:**cloud storage vendors provide access tiers to the data depending on the frequently accessed data. This saves cost as the infrequently accessed data is put in the archive tier that has lower cost for storing the data than the hot tier that is used with frequently accessed data.

Cloud Storage drawbacks:

- **Internet utilization:** one of the cloud storage cons is the need for high internet speed with high bandwidth since any interruption in the internet speed and bandwidth affects the performance of taking backups for the data and restoring it.

- **Security:** one of the concerns for any organization to move to cloud storage is that it has no control on the data on the cloud compared to the data on premises. **Conclusion:** Moving to the cloud storage increases day by day since it has less drawbacks compared to its benefits as it reduce cost and enable high availability and durability.

3.7 Azure Storage

In this section we will focus on one of the services that Microsoft cloud service provider(see section 1.6) offers which is Azure Storage. Azure Storage enables different types of storage services such as storing for a file system service for the cloud,disk storage for Azure virtual machines (VMs),data objects, a messaging store for reliable messaging, and a NoSQL store. These services characterized by the following:

- **Accessibility:**You can access the data in Azure Storage from anywhere in the world over HTTP or HTTPS.Client libraries with different languages is offered to Azure Storage by microsoft such as Go, Python, .NET, Java, Node.js, PHP, Ruby, and others, in addition to REST API. Azure Storage also enables an easy way to deal with data by providing GUI interfaces such as Azure portal and Azure Storage Explorer .Azure also supports scripting in Azure PowerShell or Azure CLI.
- **Scalability:** Azure Storage offers high scalability to face the data storage and performance needs of today's applications.
- **Managed:** All the management needs for hardware maintenance, updates, and critical issues is done by Azure .
- **Secure:** Azure offers you control over who has accessed your data.Azure encrypts all the written data.
- **Durability and highly availability:** Using redundancy techniques provides durability for the data and gives high availability in case of outage scenarios by having multiple copies of the data stored in different regions.

  To store any data to Azure Storage you should first create a storage account which contains Azure Storage data(see section 3.7.1) .Before displaying the different types of Azure Storage Account we will take a whole look for the hierarchy of Azure resources. [32]

Hierarchy of Azure resources

The following figure shows the hierarchy of Azure resources that consists of the Subscription,Resource Group,Resources,Storage account and different types of data that are stored in storage account.

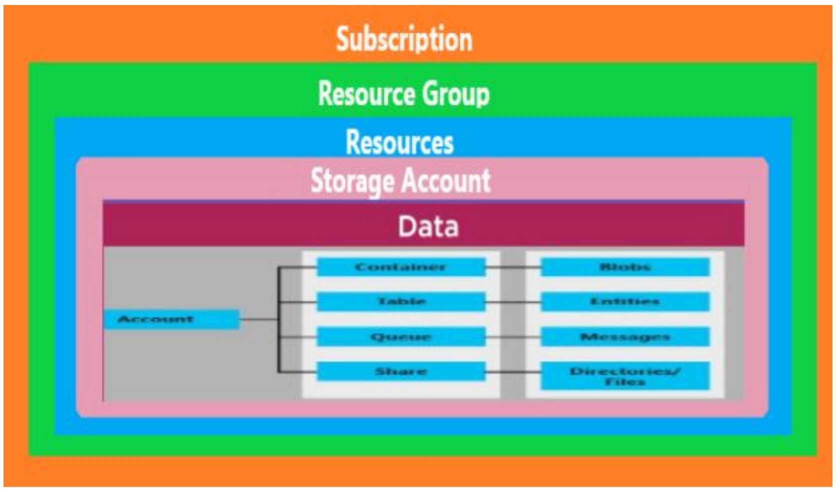


Figure 3.5: Hierarchy of Azure resources

- **Subscription:**First stage in the hierarchy of Azure resources is having a subscription on Azure.Subscription is a logical container that contains all the azure resource groups and resources.Th cost of any service is managed at the subscription level for example if you create a virtual machine and run it for a time, the cost will be managed at the subscription.
- **Management Group:**Management Group is a logical container that provides a management level above subscriptions for organizations that have many subscriptions.It enables the organization to manage the access and policies of these subscriptions by creating a management group contains subscriptions and applying to it the conditions and policies needed,all the subscriptions in this management group inherit the conditions applied on it.
- **Resource Group:**Resource Group is a logical container that groups resources together so they can be managed as a single entity.
- **Resources:**Resources are any item that can be managed by Azure such as virtual machines,storage accounts and virtual networks.
- **Storage Account:**Storage Account holds all of Azure Storage data objects: blobs, files, queues, tables, and disks(see section Azure storage services .Data in Azure storage account is secure,durable and highly available and scalable.

3.7.1 Azure Storage Account

Azure Storage Account is the parent that all the Azure Data objects as files,blob,tables,disks and queues(see section 3.9) are created in it.Azure Storage Account can be accessed from anywhere through HTTP or HTTPS and storage account has a unique namespace that can be repeated.All the data in the Azure Storage Account are secure,durable and highly available and scalable.

Azure Storage Account has five types of storage account each type has its own features and price.Theses five types are General Purpose version 1(GPv1),General Purpose version 2(GPv2),Blob Storage,Block Blob Storage and File Storage. [33]

We will illustrate each type and the difference between them.

3.7.2 General Purpose version 1(GPv1):

GPv1 supports all Azure service which are blob(all types),table,files,queues and disks(see section 3.7.1.1).GPv1 does not support the latest feature,it supports deploying with Class model so we may use GPv1 to migrate with virtual machines and virtual networks that were deployed using the Classic model.GPv1 does not support the access tiers hot,cool and archive(see section 3.9.1.3).

3.7.3 Blob Storage Account:

Blob Storage Account supports the Blob services (section 3.7.1.2) only and does not include file,table and queues.Blob account supports access tiers to enable storing blob in hot,cool and archive(see section 3.9.1.2) to reduce the cost of storing in frequently unused data.

3.7.4 General Purpose version 2(GPv2):

GPv2 supports all Azure service blob(all types),table,files,queues and disks(see section 3.9.1.3).GPv2 supports the latest features and supports access tiers: hot,cool and archive(see section 3.7.1.3).GPv2 is lower in cost than GPv1.Microsoft recommends to use GPv2 and it easy to upgrade from GPV1 or Blob Storage account to GPv2. GPv1 and GPv2 support their services on standard performance backed by hard disk drives. And also GPv1 and GPv2 offer the premium performance tier which is backed by SSD(Solid State Drives). In 2019, new storage two account types were launched: Block Blob Storage and File Storage that allow storing blobs and files with premium performance characteristics.

3.7.5 Block Blob Storage Account:

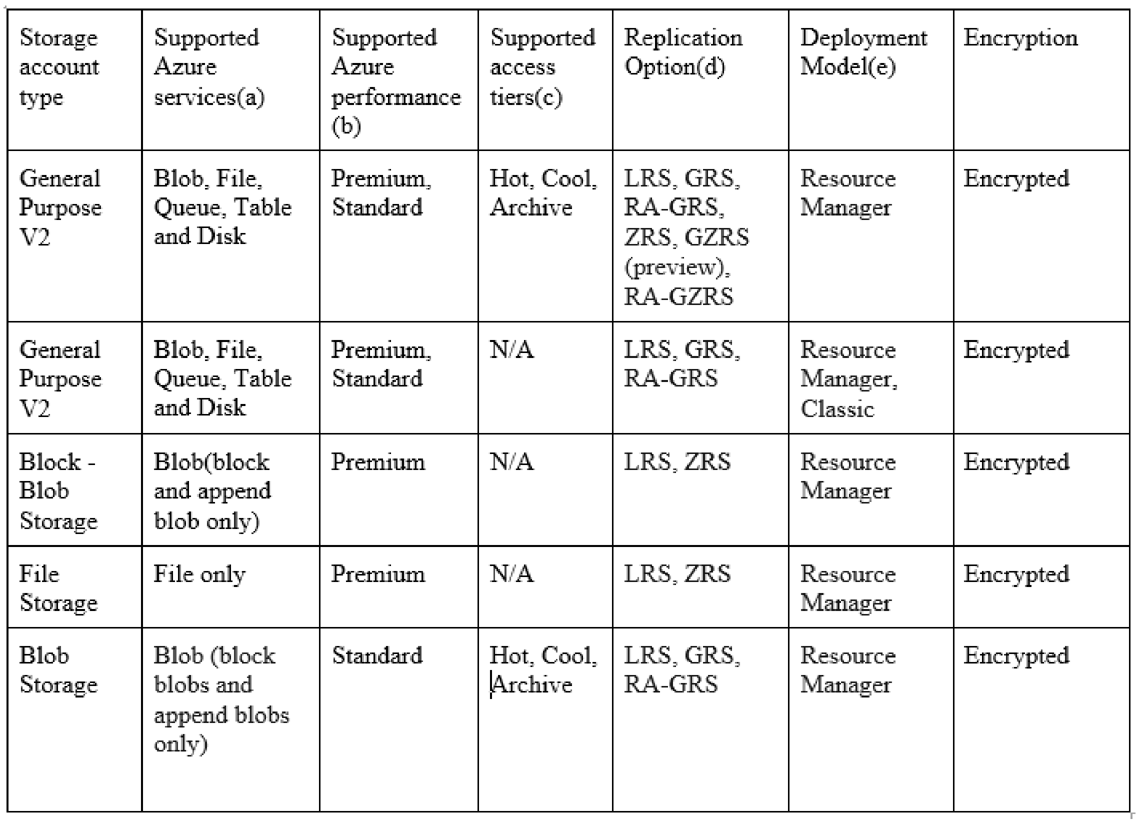
Block Blob Storage account supports the Blob azure service only does.It enables storing both Block and Append blob only with premium performance characteristics which provides better performance for scenarios with high transaction rates.

3.7.6 File Storage Account:

File Storage account supports file shares backed by premium SSD.It does not support blob,table,queues or disks.File Storage Account is recommended for enterprise or high performance scale applications.

The following figure shows the different types of storage account and their capabilities:

Figure 3.6: Different Types of Azure Storage Account



Storage account type	Supported Azure services(a)	Supported Azure performance (b)	Supported access tiers(c)	Replication Option(d)	Deployment Model(e)	Encryption
General Purpose V2	Blob, File, Queue, Table and Disk	Premium, Standard	Hot, Cool, Archive	LRS, GRS, RA-GRS, ZRS, GZRS (preview), RA-GZRS	Resource Manager	Encrypted
General Purpose V2	Blob, File, Queue, Table and Disk	Premium, Standard	N/A	LRS, GRS, RA-GRS	Resource Manager, Classic	Encrypted
Block - Blob Storage	Blob(block and append blob only)	Premium	N/A	LRS, ZRS	Resource Manager	Encrypted
File Storage	File only	Premium	N/A	LRS, ZRS	Resource Manager	Encrypted
Blob Storage	Blob (block blobs and append blobs only)	Standard	Hot, Cool, Archive	LRS, GRS, RA-GRS	Resource Manager	Encrypted

[34]

1. **Azure Services:** Azure supports 5 types of services which are:Blob,File,Queue,Table and Disk(see section 2.3)

2. **Performance Tiers:**There are two performance Standard and Premium.

   - **Standard Performance:**backed by HDD and it is suitable for high capacity and high throughput.
   - **Premium Performance:**backed by SDD and is suitable for high transaction rates.

3. **Access Tiers:**There are three access tiers: hot,cool and archive(see section 3.3.1.2) which save the cost of storing infrequently unused data.

4. **Replication Options:**One of the benefit of Azure Storage is the durability and availability of the data by storing multiple copies of data in different locations. There are four types of replication LRS,ZRS,GRS/RA-GRS and GZRS/RA-GZRS(see section3.2)

5. **Deployment Model:**There are two deployment models that are provided by Azure which are Classic deployment model and Resource manager.

   - **Classic Model:** In this model each resource is managed independently,related resources can not be grouped together.To create or delete a solution, you have to deal with each resource individually and you can not apply access control policies for the related resources.
   - **Resource Manager:** This model provides grouping related resources together and so you can manage,deploy and monitor all related resources as a group.In this model you can apply the access control policies for the group and it will apply automatically to each resource in this group.

3.7.7 Azure Storage Redundancy and backup

Azure Storage offers high durability and availability by storing multiple copies of data that data can be protected from natural disaster action and from any other event such as power outages,network outages or transient hardware failures. Before talking about the different redundancy options there are keywords that should be known to easy understand Azure redundancy options [35]:

- **Region:**A group of data centers installed within a specific defined perimeter and associated through a dedicated low-latency network.
- **Geography:**An area of the world that contains a minimum one Azure region,this gives the customer higher availability and durability for their data in case of complete region failure.
- **Availability Zone:**Unique physical locations within a region where each availability zone consists of one or more data centers installed in it with separated power,networking and cooling.

The figure below shows the architecture Zones within Azure region:

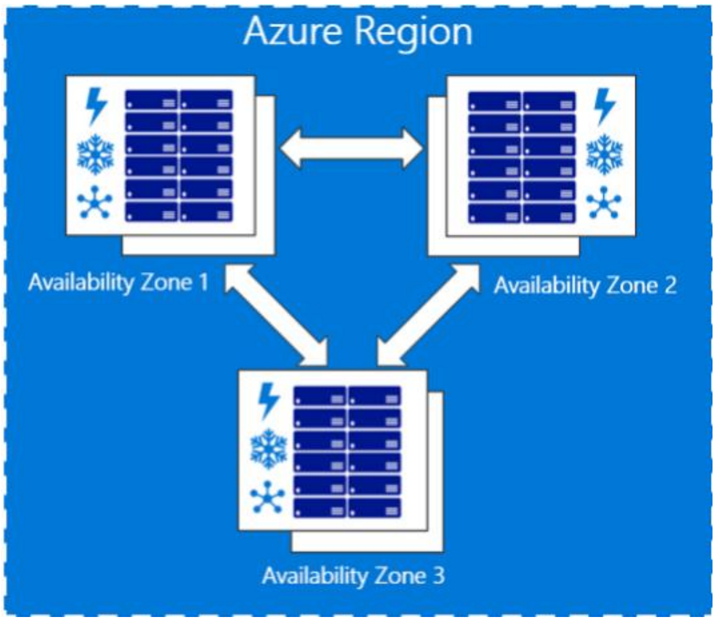


Figure 3.7: Availability Zones within Azure Region
[36]

Azure Storage offers two redundancy options:Redundancy in the primary region and Redundancy in the secondary region.

3.7.8 Redundancy in the primary region

The data is stored three times in the primary region.There are two options offered by Azure Storage which are: Local Redundancy Storage (LRS) and Zone Redundancy Storage(ZRS).

- **Local Redundancy Storage (LRS):** In LRS you data is replicated three times within a single physical location in the primary region.LRS supports at least 99.999999999 percent (11 nines) durability of the data over a given year.Storage account that uses LRS the write operation happens synchronously and after the data is been written the write operation returns successfully.Although LRS is the lowest redundancy option in cost but it also the least durability compared to other options.In LRS you data can be protected against server rack and drive failures but you data can not be protected if fire or flooding happens within the data center and so all replicated data of a storage account may be lost or unrecoverable.
- **Zone Redundancy Storage(ZRS):** In ZRS you data is replicated three times across three Azure availability zones in the primary region.Each availability zone is a separate physical location with independent power, cooling, and networking. .ZRS supports at least 99.9999999999 percent (12 nines) durability of the data over a given year.Storage account that uses ZRS the write operation happens synchronously and after the data is been written the write operation returns successfully.ZRS can not protect your data if a regional disaster happens and so

Microsoft recommends to replicate your data in another secondary region beside replicating your data in the primary region is called Geo-zone-redundant storage (GZRS).

3.7.9 Redundancy in the secondary region

The data is stored in a secondary region beside the primary region which is far away hundreds of miles. This is used with the applications that need high availability.This gives high durability for the data even in the case of complete regional outage or a disaster happens in the primary region. There are two options offered by Azure Storage which are: **Geo-redundant storage (GRS)** and **Geo-zone-redundant storage (GZRS)**.The difference between GRS and GZRS is the way of data replication in the primary region.Data replicated in secondary three times using LRS that protects the data against hardware failures.

- **Geo-redundant storage (GRS):** In GRS,the data is replicated three times within a single physical location in the primary region using LRS then the data is asynchronously copied to a single physical location in a secondary region far away hundreds of miles of the primary region.GRS supports at least 99.99999999999999 percent (16 9's) durability of the data over a given year. A write operation is first done to the primary location and replicated using LRS. The update is then replicated asynchronously to the secondary region. The written data in the secondary location is also replicated using LRS within that location.
- **Geo-zone-redundant storage(GZRS):** In GZRS,three times across three Azure availability zones in the primary region then the data is asynchronously copied to a secondary geographic region to protect the data from regional disasters.GZRS supports at least 99.99999999999999 percent (16 9's) durability of the data over a given year.
- **Read access to data in the secondary region(RA-):** Geo-redundant storage is associated with GRS or GZRS.In RA-GRS or RA-GZRS the replicated data in the secondary region is available to be read only.The data in the secondary region can be read even if the primary region becomes unavailable this protects the data against regional outages. Another benefit for Read Access,that if the storage performance of your application is interrupted, you have the chance to redirect some the read operations to the secondary URL which decreases the IO per second on the primary location.The figure below displays how Read Access works as secondary read location with the primary location.

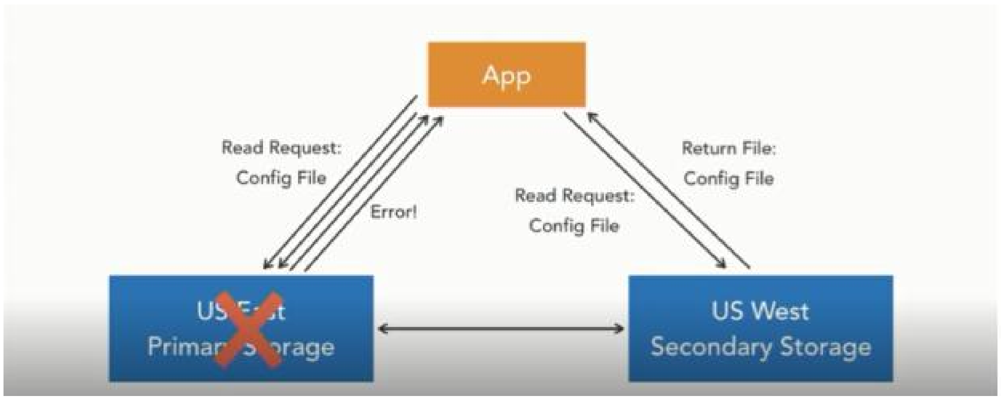
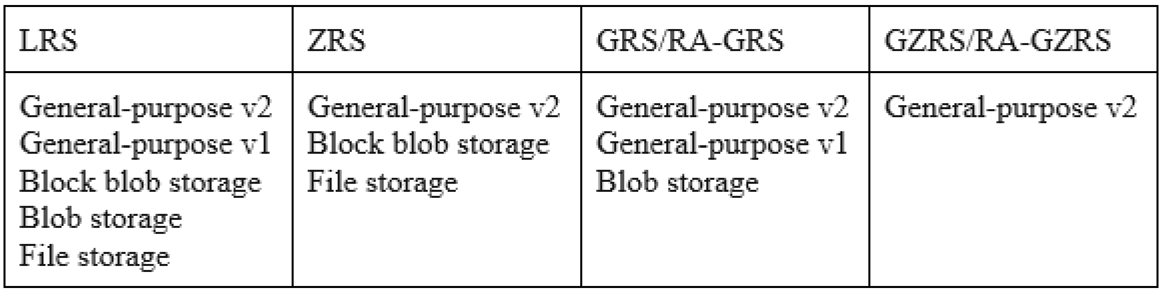


Figure 3.8: Read Access workflow
[36]

The following figure shows the redundancy options offered by the different types of storage accounts

Figure 3.9: the redundancy options supported by storage accounts



LRS	ZRS	GRS/RA-GRS	GZRS/RA-GZRS
General-purpose v2 General-purpose v1 Block blob storage Blob storage File storage	General-purpose v2 Block blob storage File storage	General-purpose v2 General-purpose v1 Blob storage	General-purpose v2

[37]

3.7.10 Azure storage services

Microsoft offers several options to store data on the cloud. Each option has its unique purpose for serving different business needs. The Azure Storage offers the following data services:

- **Azure Blobs:** Storing unstructured data as text and binary data.
- **Azure Files:** Storing file shares.
- **Azure Queues:** For messaging between applications.
- **Azure Tables:**Storing structured NoSQL data.
- **Azure Disks:**Storing VMs.

3.7.11 Azure Blob Storage

Blob storage is Microsoft Azure's service for storing Binary Large Objects or BLOBs which are composed of unstructured data. Unstructured data is data that doesn't have a specific data model or definition such as text, images, videos, along with their metadata. Blob storage is created to:

- Stream videos and audio.
- Provide documents and images directly to a browser.
- Enable writing to log files.
- Store files for distributed access.

Objects in Blob storage can be accessed by users or client applications via HTTP/HTTPS, from anywhere in the world. Objects in Blob storage can be accessed via the Azure Storage REST API, Azure PowerShell, Azure CLI, or an Azure Storage client library. Client libraries are available for different languages, including: .NET, Java, Node.js, Python, Go, PHP, Ruby.

3.7.11.1 Blob Storage resources

Blob Storage contains three types of resources:Storage Account,Container and Blob. The following figure shows Blob storage resources and the connection between them.

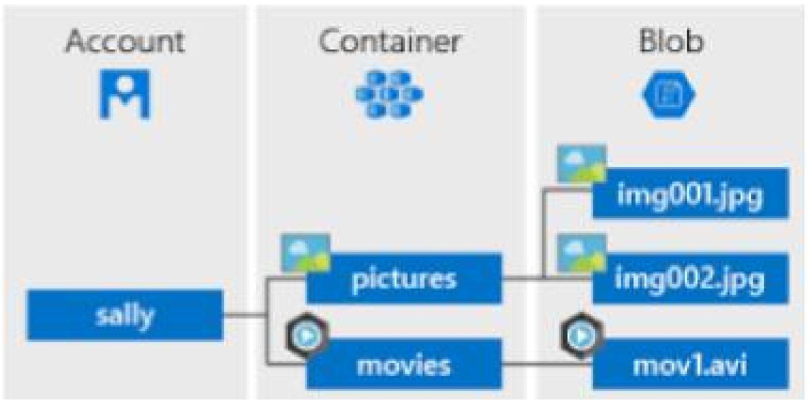


Figure 3.10: Blob Resources
[38]

- **Storage Account:** Supports a unique namespace in Azure for your data. Every object that you store in Azure Storage has an address that includes your unique account name. The base address for any object in the storage is a combination between the Azure Storage blob and the account name. For example, if the storage account name is ¡mystorage¿ so the default endpoint for Blob storage is http://mystorageaccount.blob.core.windows.net
- **Containers:**Manages a set of blobs.A container can manage an unlimited number of blobs and A storage account can contain an unlimited number of containers.
- **Blob:** Provides three types Block Blob,Append Blob and Page Blob.

3.7.11.2 Blob Storage Types

There are three types of blobs [39]: block blobs, append blobs, and page blobs. When creating the blob,first you choose the blob type. After creating the Blob, its type cannot be changed. All blobs reflect changes immediately. Any blob can be duplicated in a snapshot.

- **Block Blob storage:** Block blobs are blobs that are broken down into smaller units called blocks. Each block has an ID, and you can create, read, or modify individual blocks within a block blob. A block blob can maximum contain up to 50.000 blocks with different size. In 2019 a new version was released with maximum block size up to 4000 MiB and maximum blob size up to Approximately 190.7 TiB (4000 MiB X 50,000 blocks). Block blobs contain features that help you handle large files over networks, you can upload multiple blocks in parallel to decrease upload time. Blocks can be uploaded in any order, and their sequence can be determined in the final block list commitment step. Each block can include an MD5 hash to verify the transfer, so you can track upload progress and re-send blocks as needed. Maximum number of blocks in the blob are 50, 000 blocks, so you need to choose your block size accordingly in order to maximize the allowable size for large files. Breaking down a blob into small blocks can help with low-bandwidth internet connections, so it's good to know these options when planning your solution.

- **Append Blob Storage:** Append blobs are also composed of blocks, but append blobs are optimized for adding a blob only. You can only add blocks to the end of the blob. Updating and deleting existing blocks isn't available. Append blob is ideal for log and audit files because it ensures that no one can modify the file, so that might help with regulatory compliance, for example. But of course, the person with the right permissions could still delete the file. Each block in an append blob can be a different size, up to a maximum of 4 MiB, and an append blob can contain up to 50,000 blocks. The maximum size of an append blob is slightly more than 195 GiB (4 MiB X 50,000 blocks).

- **Page Blob Storage:** Page blobs are used for random read/write operations. Page blobs are a collection of 512-byte pages with a maximum size of Page blob is 8TB. such as VHD's(Virtual Hard disk) .Page blob are used to store VHD. Azure supports two types of disk storage: premium and standard. Premium Storage is the most recent storage option designed for Azure virtual machine workloads that provides users with data storage on SSD(Solid State Drive) for better IO performance and low latency. Premium Storage supports only the Page blob.While Standar Storage uses HHD suitable with operations .

3.7.11.3 Blob Storage Access Tiers

Azure Storage offers three types for accessing the Blob data:hot,cool and archive.These tiers reduce the cost of storing infrequently unused data [40] :

- **Hot Access Tier:** This tier is suitable for frequent access of data in the storage account. In the hot tier accessing the data is less in cost than storing the data. When creating a storage account the hot tier is the default choice.

- **Cool Access Tier:** This tier is suitable for the data that is infrequently accessed and the data is stored for at least 30 days.Accessing the data in cool tier is more expensive than hot tier while Storing data in the cool tier is more cost-effective.

- **Archive Access Tier:** This tier is offered only for block blobs. The archive tier is suitable for data that can afford many hours to be retrieved.The data remains in the archive tier for at least 180 days.Storing the data in the archive tier is the most cost-effective while accessing the data in the archive tier is more expensive than the hot or cool tiers.

3.7.11.4 Blob Storage Life cycle

The data in the blob storage passes through three stages: hot,cool and archive. The following figure shows Blob life cycle and the maximum time for remaining the data in each tier.

1. The data remain in the hot tier for frequent read and write operation for 48 hours. In this stage the cost of accessing the data is less than storing the data.

2. After the 48 hours, the data that is infrequently accessed goes to the cool tiers and remains at least 30 days and even if you delete the data from the cool tier before 3o days you will also pay for it.

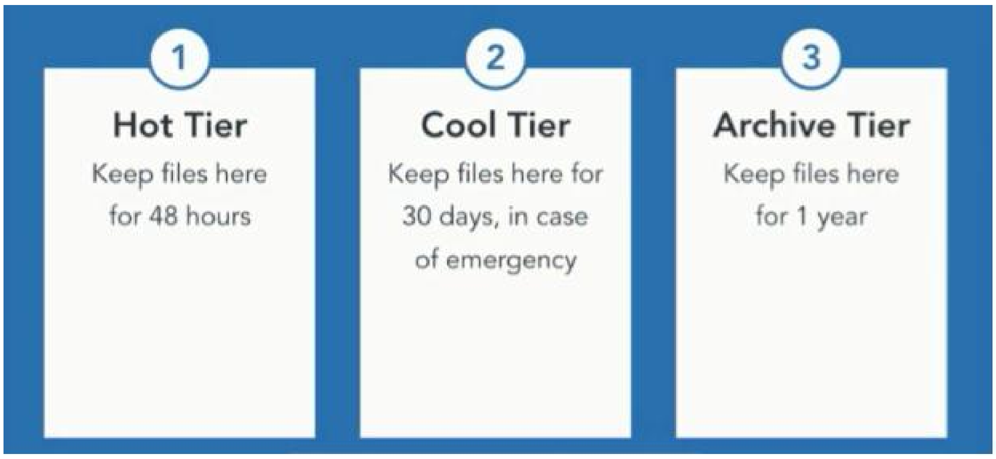

1
Hot Tier
Keep files here for 48 hours
2
Cool Tier
Keep files here for 30 days, in case of emergency
3
Archive Tier
Keep files here for 1 year

Figure 3.11: Data Life Cycle
[38]

3. After 30 days,the data goes to the last stage which is the archive tier and remains at least 180 days and even if you delete the data from the archive tier before180 days you will also pay for it.you can keep the data in the archive tier for 1 year.

3.7.12 Azure Queue Storage

Azure Queue storage delivers messaging between applications.When designing applications for scale,to enable the components to scale independently ,they are decoupled.Queue storage offers asynchronous messaging between application components even if they are running on-premises server, in the cloud,on desktop or mobile device. [41]

Stored messages in Azure Queue storage can be accessed from anywhere in the using HTTP or HTTPS. The size of a single queue message can be up to 64 KB, and a queue can contain massive numbers of messages, up to the capacity limitation of a storage account. Queue storage is used for the creation of a backlog of the work to be processed asynchronously.The messages in the Queue Storage is put in order with the concept FIFO(First In,First Out) and when try to get one of these messages out of the queue,it asks you if you want to remove the first element which ensures the intention of a FIFO (First In First Out) queue.

Queue service concepts
Queue Service consists of URL format,Storage Account,Queues and Messages. The following figure shows Queue service and the connection between them.

- **URL format:**Queues are displayed in Azure account in the following URL format: http://¡storage account¿.queue.core.windows.net/¡queue¿ So the queue in the diagram will be displayed as the following:
  http://myaccount.queue.core.windows.net/incoming-orders

- **Storage Account:** Parent that contains all the data and the accessed way to any data. Queue: Queue includes all the messages .A Queue contains a set of messages.

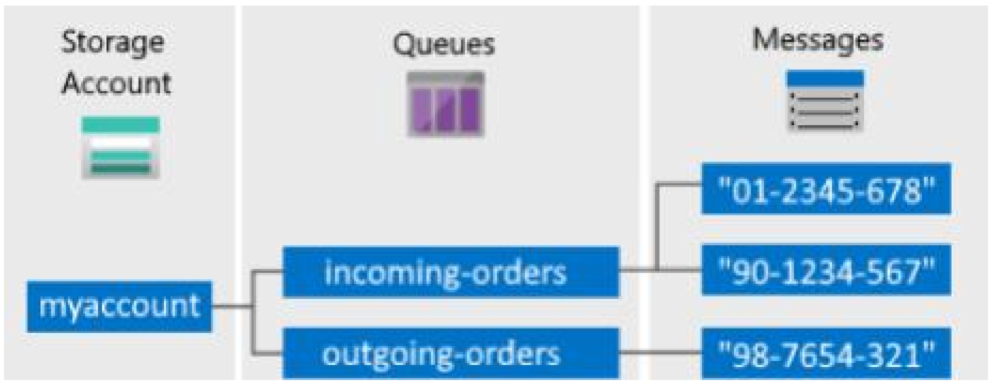


Figure 3.12: Queue Service components
[38]

- **Message:**The size of the message can be up to 64 KB and the message can be in any format. The message can remain in the queue for a maximum 7 days. The maximum time-to-live for a message in the queue can be any positive number, or -1 indicates that the message doesn't expire. The default time-to-live for a message in the queue is 7 days.

3.7.13 Azure Files

Standard Server Message Block (SMB) protocol is used to access Azure. Azure file shares can be mounted at the same time by cloud or on-premises deployments of Windows, Linux, and macOS. [42]
Additionally,as shown in the following figure, Azure file requires a storage account.

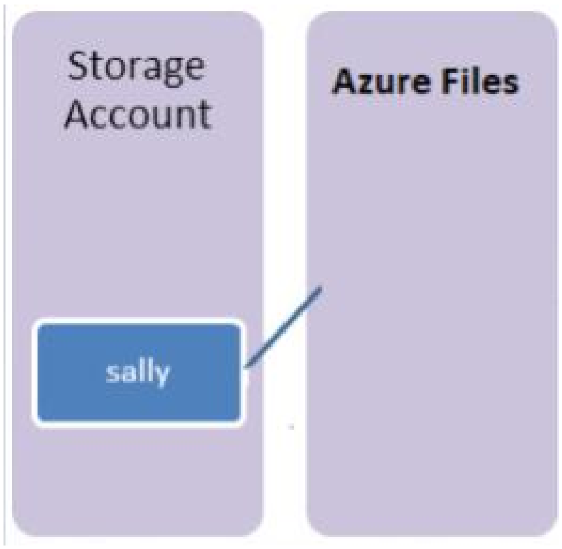


Figure 3.13: File Share
[38]

Advantages of Azure file

Azure file shares can be used to:

- **Replace or supplement on-premises file servers:** Azure Files might be used to substitute traditional on-premises file servers or NAS devices. Popular operating systems such as Windows, macOS, and Linux can directly mount Azure file shares in any part of the world. Azure file shares can also be replicated with Azure File Sync to Windows Servers, either on-premises or in the cloud, for performance and distributed caching of the data where it's being used. With the recent release of Azure Files AD Authentication.
- **"Lift and shift" applications:** Azure Files makes it easy to "lift and shift" applications to the cloud that is programmed to write to a file share to store file application or user data. Azure Files supports both the "classic" lift and shift scenario, where both the application and its data are moved to Azure, and the "hybrid" lift and shift scenario, where the application data is moved to Azure Files, and the application continues to run on-premises.
- **Simplify cloud development:** Azure Files can also be used in various ways to ease migration to new cloud development projects. For example:
- **Shared application settings:** A well-known practice for distributed applications is to save configuration files in a centralized location so that many application instances can access. Application instances can get their configuration through the File REST API, and humans can access them as needed by mounting the SMB share locally.
- **Diagnostic share:** An Azure file share is a good place for cloud applications to store their logs, metrics, and crash dumps. The application instances can write through the File REST API, and developers can access them by mounting the file share on their local machine. Offering a great flexibility, as developers can adapt cloud development without having to leave any existing tooling they know and love.
- **Dev/Test/Debug:** When developers or administrators are working on VMs in the cloud, they usually require a group of tools or utilities. Copying such utilities and tools to each VM can be a time consuming practice. By mounting an Azure file share locally on the VMs, a developer and administrator can quickly access their tools and utilities, no copying required. [42]

The figure below shows how azure file is mounted to a computer.

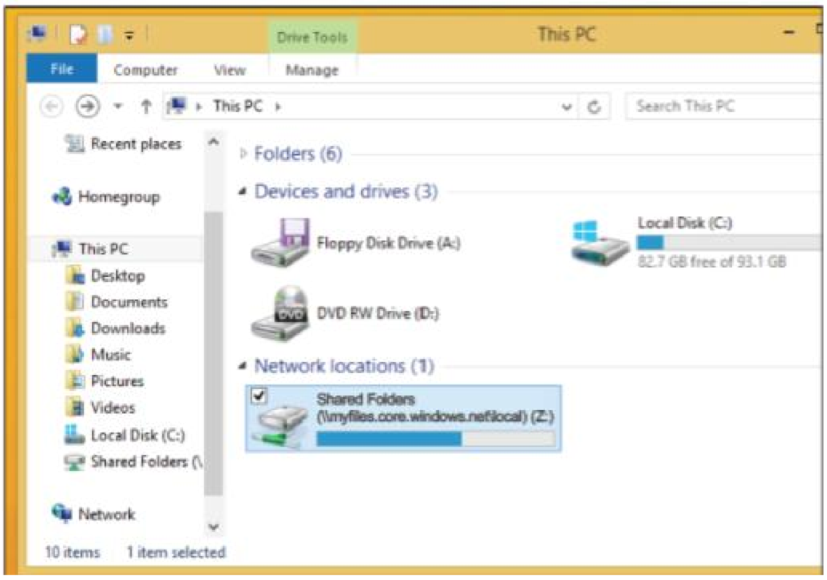


Figure 3.14: Mounted file share
[38]

3.7.14 The Azure table Service

The Azure Table storage and the Azure Cosmos DB are services that store structured NoSQL data in the cloud, using a key/attribute store with a "schemless" design. Because Table storage and Azure Cosmos DB are "schema less", it's easy to scale storage with the growing data associated with the requirements of your application. A lot of applications can interact with tables very quickly and in a cost effective way .And is more cheaper than traditional SQL for the same size of data.the following figure shows that a table requires a storage account then each table consists of group of entities.

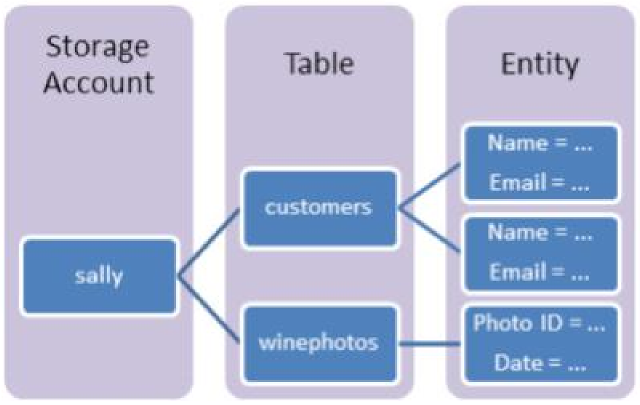


Figure 3.15: Table in Azure
[43]

You can use the Table storage or the Azure Cosmos DB to manipulate and save flexible data sets like user data for web applications, address books, device information, or other types of metadata your service requires. You can save any number of entities

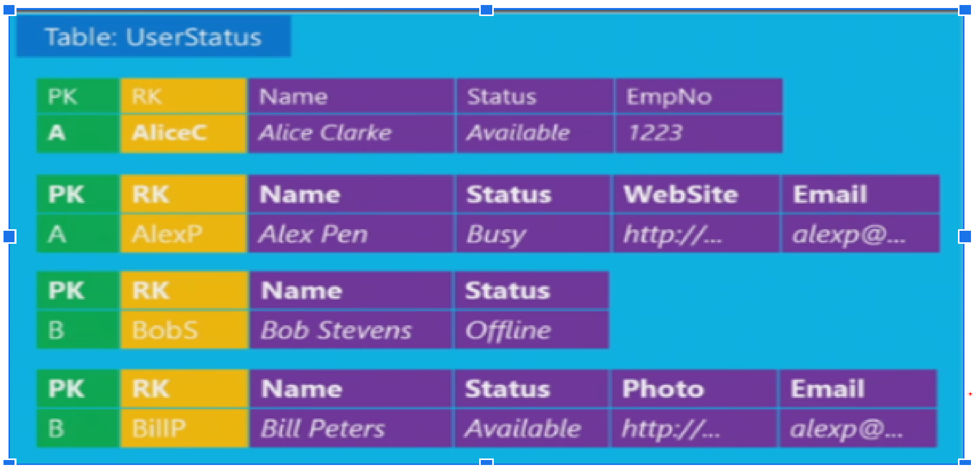


Table: UserStatus

PK	RK	Name	Status	EmpNo
A	AliceC	Alice Clarke	Available	1223

PK	RK	Name	Status	WebSite	Email
A	AlexP	Alex Pen	Busy	http://...	alexp@...

PK	RK	Name	Status
B	BobS	Bob Stevens	Offline

PK	RK	Name	Status	Photo	Email
B	BillP	Bill Peters	Available	http://...	alexp@...

Figure 3.16: Table in Azure
[43]

in a table, and a storage account can include any number of tables, up to the capacity limit of the storage account. [44]

3.7.14.1 Table Structure

Each table on Azure Table Storage is composed of five main components:

- **Entities:** are like rows (typical records) on relational databases. Partition keys: Strings that contain no more than 1 KB of data. It can aggregate one or more entities in a table, grouping them into blocks of entities that have the same ID.
- **Row keys:** Strings that contain no more than 1 KB of data. They are used to identify rows inside tables uniquely.
- **Timestamps:** Properties that store when entities were inserted or updated.
- **Partitions:**Collections of entities in a table. Partitions are identified by their partition key to group them like several tables but inside one object.the following figure summarizes the structure of table.

3.7.14.2 System Properties

An entity always has the following system properties:

- Partition Key property
- Row Key property
- Timestamp property

These system properties are automatically inserted for each entity in a table. The names of these properties are reserved and cannot be changed. One of the developer role is

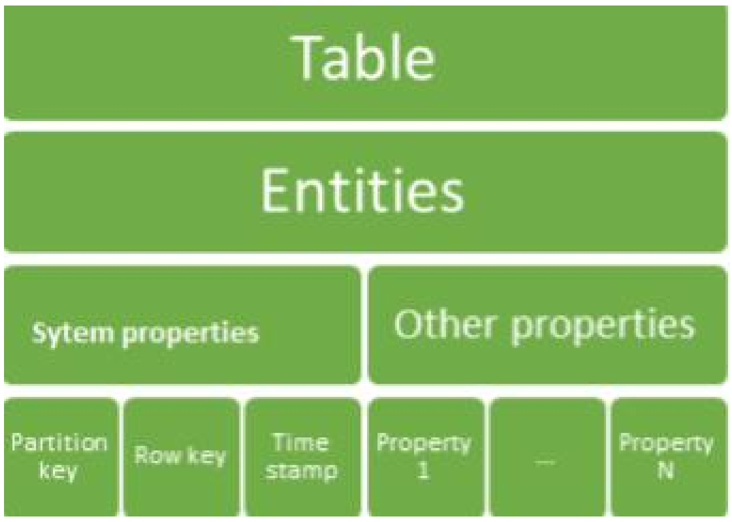
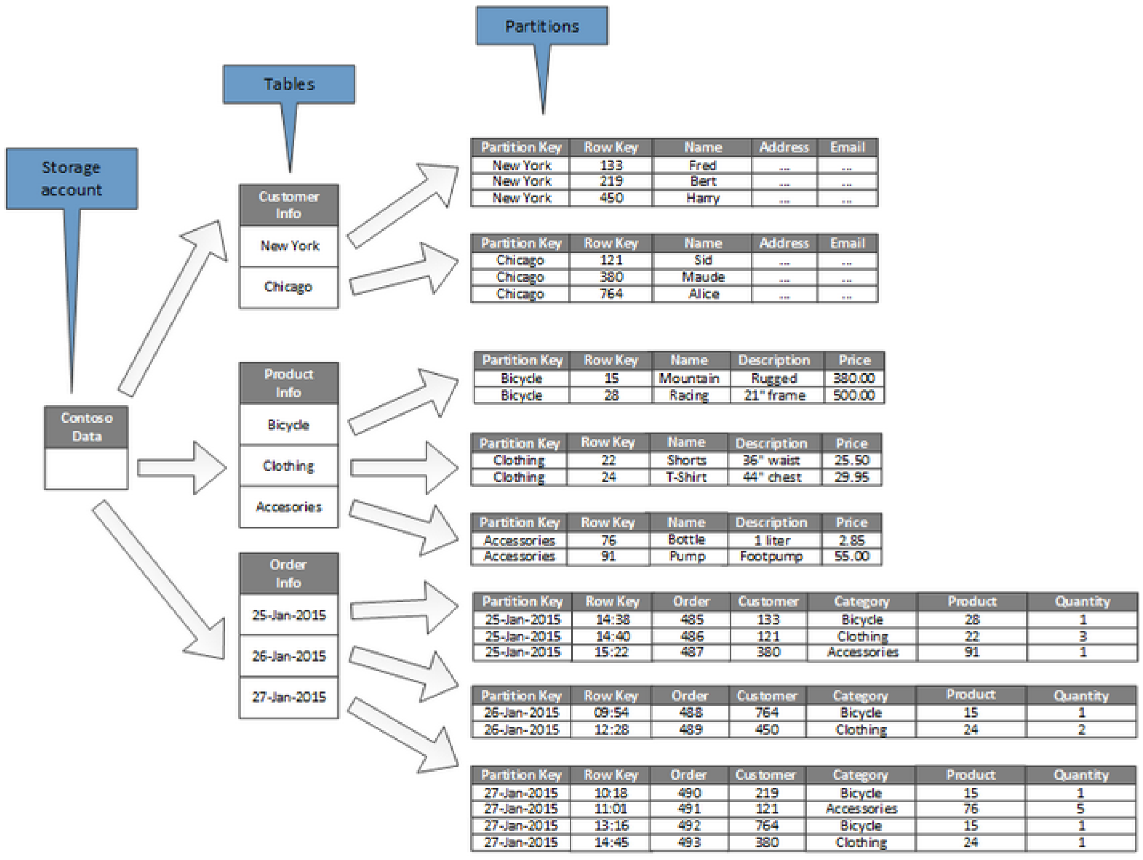


Figure 3.17: Table Structure
[38]

defining, inserting and updating the values of Partition Key and Row Key.It's the server function to generate the value of Timestamp, which cannot be altered.the following table represents tables with partitions.



Partition Key	Row Key	Name	Address	Email
New York	133	Fred	...	...
New York	219	Bert	...	...
New York	450	Harry	...	...

Partition Key	Row Key	Name	Address	Email
Chicago	121	Sid	...	...
Chicago	380	Maude	...	...
Chicago	764	Alice	...	...

Partition Key	Row Key	Name	Description	Price
Bicycle	15	Mountain	Rugged	380.00
Bicycle	28	Racing	21" frame	500.00

Partition Key	Row Key	Name	Description	Price
Clothing	22	Shorts	36" waist	25.50
Clothing	24	T-Shirt	44" chest	29.95

Partition Key	Row Key	Name	Description	Price
Accessories	76	Bottle	1 liter	2.85
Accessories	91	Pump	Footpump	55.00

Partition Key	Row Key	Order	Customer	Category	Product	Quantity
25-Jan-2015	14:38	485	133	Bicycle	28	1
25-Jan-2015	14:40	486	121	Clothing	22	3
25-Jan-2015	15:22	487	380	Accessories	91	1

Partition Key	Row Key	Order	Customer	Category	Product	Quantity
26-Jan-2015	09:54	488	764	Bicycle	15	1
26-Jan-2015	12:28	489	450	Clothing	24	2

Partition Key	Row Key	Order	Customer	Category	Product	Quantity
27-Jan-2015	10:18	490	219	Bicycle	15	1
27-Jan-2015	11:01	491	121	Accessories	76	5
27-Jan-2015	13:16	492	764	Bicycle	15	1
27-Jan-2015	14:45	493	380	Clothing	24	1

Figure 3.18: Table with partition
[45]

3.7.14.3 Partition Key Property

Partitioning of tables is a technique used to apply load balancing across storage nodes. Partitions are the components that categorize entities based on the value of the partition key. A partition is a successive range of entities owing the same partition key value. The partition key is a unique identifier for the partition within a specific table, defined by the Partition Key property. The partition key builds the first part of an entity's primary key. The partition key may be a string value up to 1 KB in size. You must include the Partition Key property in every insert, update, and delete operation.

3.7.14.4 Row Key Property

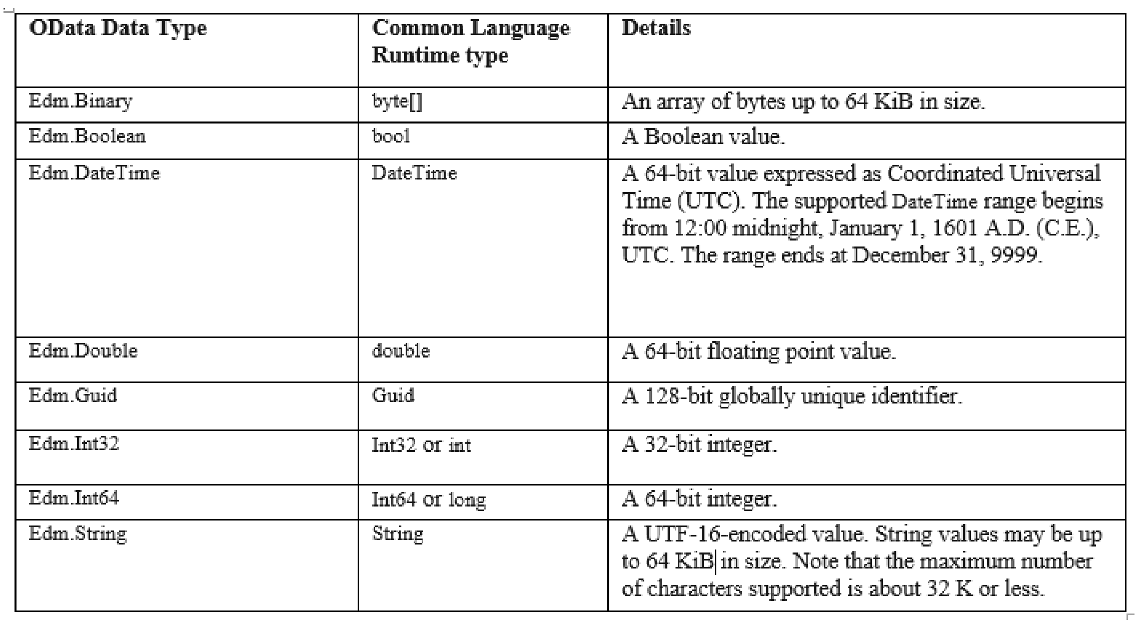
The second part of the primary key is the row key. The row key is a unique identifier for an entity inside a partition. Both the Partition Key and Row Key uniquely define every entity within a table. The row key is a string value that may be up to 1 KiB in size. You must include the Row Key property in every insert, update, and delete operation.

3.7.14.5 Timestamp Property

The Timestamp property is a Date Time value that is managed by the server to record the time an entity was last modified. Ultimate Synchronization is internally managed by The Table service through the Timestamp property. Timestamp value is an increasing value when the entity is modified, the value of Timestamp increases for that entity. This property should not be set on insert or update operations (the value will be omitted).

3.7.14.6 Property Types

The Table service offers a subset of data types described by the OData Protocol Specification. The following table shows the supported property types for the Table service:



OData Data Type	Common Language Runtime type	Details
Edm.Binary	byte[]	An array of bytes up to 64 KiB in size.
Edm.Boolean	bool	A Boolean value.
Edm.DateTime	DateTime	A 64-bit value expressed as Coordinated Universal Time (UTC). The supported DateTime range begins from 12:00 midnight, January 1, 1601 A.D. (C.E.), UTC. The range ends at December 31, 9999.
Edm.Double	double	A 64-bit floating point value.
Edm.Guid	Guid	A 128-bit globally unique identifier.
Edm.Int32	Int32 or int	A 32-bit integer.
Edm.Int64	Int64 or long	A 64-bit integer.
Edm.String	String	A UTF-16-encoded value. String values may be up to 64 KiB| in size. Note that the maximum number of characters supported is about 32 K or less.

Figure 3.19: Property type table
[46]

By default a property is created as type String, in case you assign a different type. To explicitly type a property, specify its data type by using the appropriate OData data type for an Insert Entity or Update Entity operation. Note, The Table service does not persist null values for properties. When querying entities, the above property types are all non-nullable. When writing entities, the above property types are all nullable, and any property with a null value is manipulated as if the entity did not have that property.

3.7.14.7 Rules for naming the tables

Table names must conform to these rules:

1. Table names must be unique within an account.
2. Table names may contain only alphanumeric characters.
3. Table names cannot begin with a numeric character.
4. Table names are case-insensitive.
5. Table names must be from 3 to 63 characters long.
6. Some table names are reserved, including ”tables”. Attempting to create a table with a reserved table name returns error code 404 (Bad Request).

Table names preserve the case with which they were created, but are case-insensitive when used.

3.7.14.8 Property Limitations

An entity can have up to 255 properties, including 3 system properties described in the following section. Therefore, the user may include up to 252 custom properties, in addition to the 3 system properties. The combined size of all data in an entity's properties cannot exceed 1 MB.

Characters Disallowed in Key Fields

The following characters are not allowed in values for the PartitionKey and RowKey properties:

- The forward slash (/) character
- The backslash (character
- The number sign () character
- The question mark (?) character
- Control characters from U+0000 to U+001F, including:
  - The horizontal tab ()͡ character
  - The linefeed () character
  - The carriage return ()̊ character
- Control characters from U+007F to U+009F

3.7.14.9 Global data distribution with Azure Cosmos DB - overview

Nowadays users require high response, continually up and running application In order to get low latency and high availability, the requirement of these application instances to be created in data centers that are near to the customers' location. These applications are installed in multiple data centers and are called globally distributed. Globally distributed applications need a globally distributed database and have replicas in different places in the world so that the application can run the copy of the data that's close to its users. [47]

Azure Cosmos DB is a globally distributed database service that's made to offer low latency, elastic scalability of throughput, well-defined semantics for data consistency, and high availability. In brief, if your application requires fast response time in any place in the world, and the application must be up and running all the time, and needs unlimited and elastic scalability of throughput and storage, your best choice is your application on Azure Cosmos DB.

The option is yours if you want to make your application database to be globally distributed and available in any of the Azure regions. Put the application data near the location of your customers to have a low latency. Selecting the ultimate regions relies on many factors as locations of customers and the world-wide reach of your application. Cosmos DB transparently makes a replica of the data to all the regions related with your Cosmos account. It provides a single system image of your globally distributed Azure Cosmos database and containers that your application can read and write to locally.

With Azure Cosmos DB, you can change (add or remove) the regions in association with your account whenever you need. Your application doesn't require a down time to

add or remove a region. It will be up and running and available all the time because of the multi-homing capabilities that the service provides.
The figure below shows how cosmos DB is distributed.

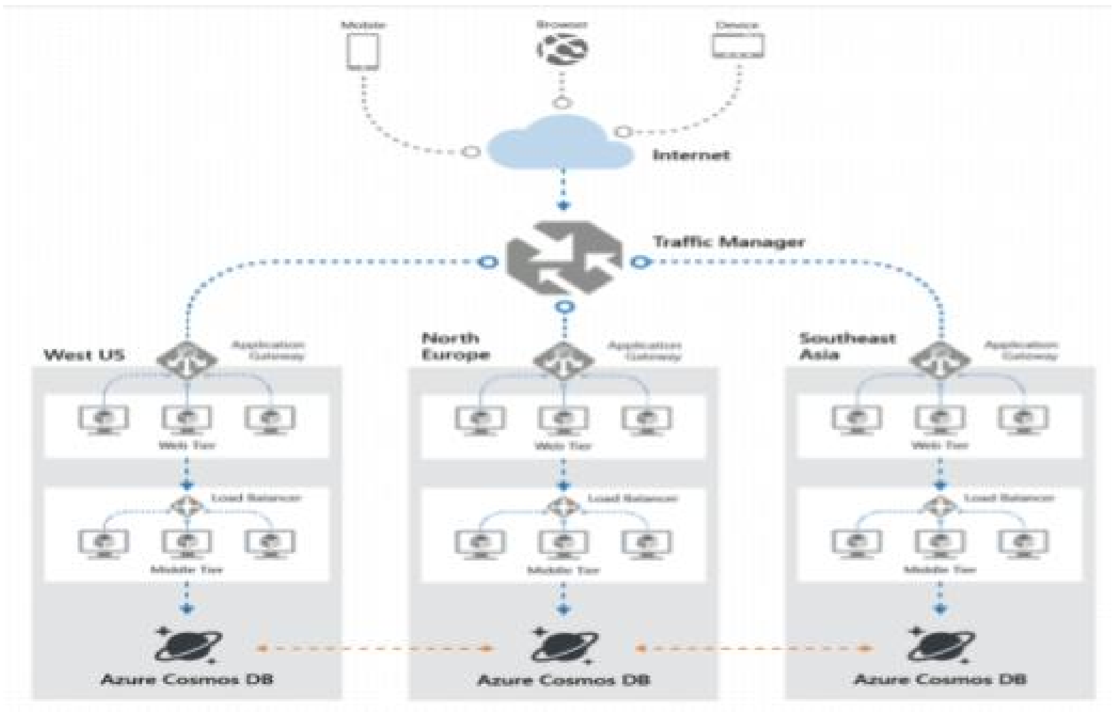


Figure 3.20: Distributed Cosmos DB
[48]

3.7.14.10 Key benefits of global distribution

1. **Global active-active apps**:The multi-master replication protocol introduced in Cosmos DB, every region supports both writes and reads. The multi-master capability also enables the following:
   - elastic write and read scalability with no limit.
   - 99.999 percent read and write availability in the world.
   - Guaranteed reads and writes served in less than 10 milliseconds at the 99th percentage.

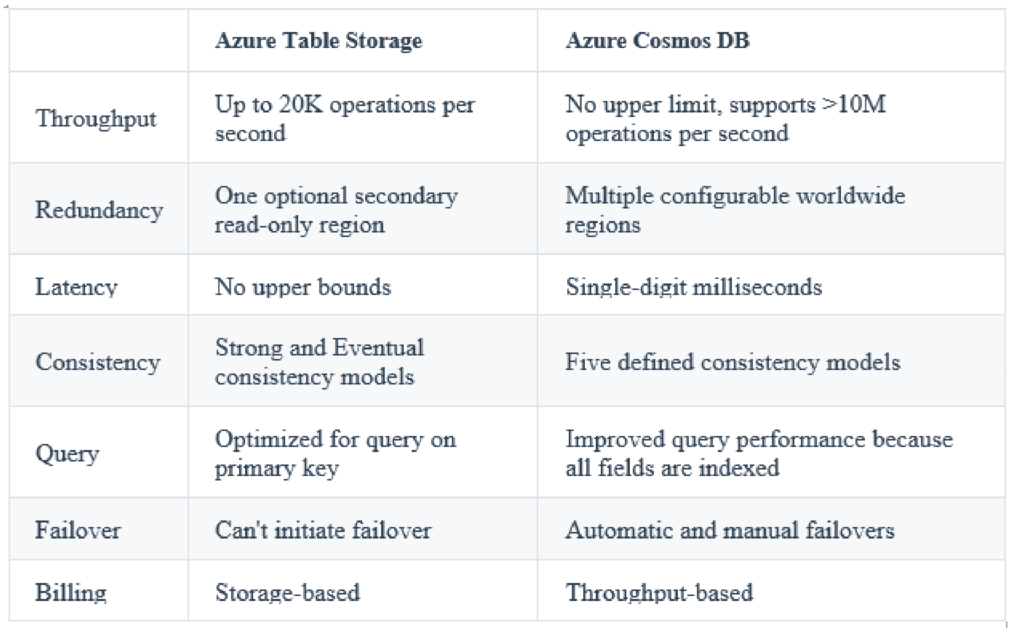
   The Azure Cosmos DB multi-homing APIs gives the ability to your application to recognize the nearest region and can send requests to that region. The nearest region can be detected without any configuration changes. Changing regions from your Azure Cosmos account, your application does not need to be redeployed or paused.
2. **Highly responsive apps:**Your application can run close to real-time reads and writes which correspond to customers' locations. Behind the scene Azure Cosmos DB manipulates the data replication between regions.

3. **Highly available apps:** Existence of database replicas in multiple regions around the world raises the availability of a database. If one region is down, other regions automatically handle application requests. Azure Cosmos DB offers 99.999 percent read and write availability for multi-region databases.

4. **Maintain business continuity during regional outages:** Azure Cosmos DB provides automatic fail over when one of the regions goes down. Azure Cosmos DB keeps to preserve its latency, availability, consistency, and throughput SLAs. To ensure that the application is highly available, Cosmos DB supports a manual fail over API to simulate a regional failure. Through this API, you can execute manual fail over.

5. **Scale read and write throughput globally:** In each region ,you can configure it to be writable and elastically scale reads and writes. The throughput that your application sets on an Azure Cosmos database or a container is guaranteed to be delivered in all regions associated with your Azure Cosmos account.

3.7.14.11 Comparison between Azure Table and Azure cosmos DB



	Azure Table Storage	Azure Cosmos DB
Throughput	Up to 20K operations per second	No upper limit, supports >10M operations per second
Redundancy	One optional secondary read-only region	Multiple configurable worldwide regions
Latency	No upper bounds	Single-digit milliseconds
Consistency	Strong and Eventual consistency models	Five defined consistency models
Query	Optimized for query on primary key	Improved query performance because all fields are indexed
Failover	Can't initiate failover	Automatic and manual failovers
Billing	Storage-based	Throughput-based

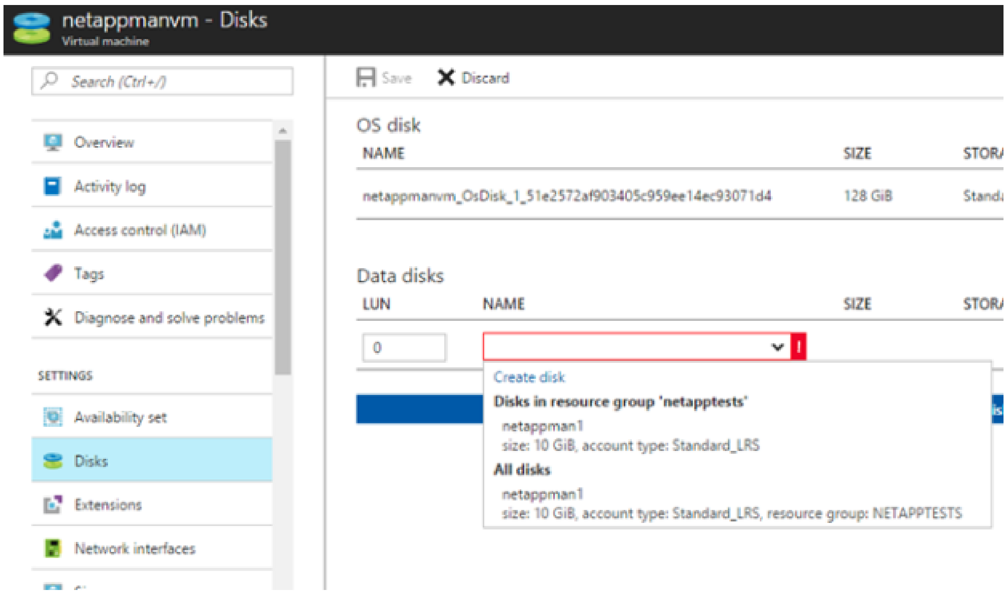
Figure 3.21: Azure Table Vs Azure cosmos DB
[49]

3.7.15 Disks in Azure

Microsoft Azure provides 2 types of virtual hard disks (VHDs): un managed and managed. What distinguishes the un managed disk that it's under the responsibility of the end user through his own storage account.(managed Disk). [50]

Azure managed disks represent a block-level storage volume that is managed and controlled by Azure in addition they are directly related to Azure virtual machines.(Azure Mnaged-Disk) Managed Disk like physical disk in an on-premises server instead it's virtualized. When you create a virtual machine you have to associate a hard disk with and that requires defining the following properties:

- **Disk size**.
- **Disk type**. The below Figure shows how to create a managed disk in Azure .



OS disk

NAME	SIZE	STOR/
netappmanvm_OsDisk_1_51e2572af903405c959ee14ec93071d4	128 GiB	Stand

Data disks

LUN	NAME	SIZE	STOR/
0			

Create disk
Disks in resource group 'netapptests'
netappman1
size: 10 GiB, account type: Standard_LRS
All disks
netappman1
size: 10 GiB, account type: Standard_LRS, resource group: NETAPPTESTS

Figure 3.22: Create Managed Disk
[51]

After the managed disk is provisioned and Azure will manipulate these disks properly, once you have the disk Azure will take the responsibility for handling and manipulating those managed disk .The available types of disks are ultra-disks, premium solid-state drives (SSD), standard SSDs, and standard hard disk drives (HDD).

-Advantages of managed disks

- **Highly available**:Managed disks guaranteed 99.999 percent availability. This is accomplished by creating three replicas of your data, If one or even two replicas stopped working , the rest of replicas will run and perform efficiently while keeping your data preserved . This architecture is known as infrastructure as a service (IaaS) disks.
- **Simple and scalable VM deployment**: Using managed disks, you can make up to 50,000 VM disks of a type in a subscription per region, allowing you to create thousands of VMs in a single subscription.
- **Integration with availability sets**:Managed disks are associated with availability sets to guarantee that the disks of VMs in an availability set are separated from each other to avoid a single point of failure. Disks are automatically assigned to various storage scale units known as (stamps). If a stamp went down due to hardware or software failure, only the VM instances with disks on those stamps will stop running. For instance, an application running on 7 VMs which are in an Availability Set. The VMs disks are not associated with the same stamp, so if one stamp fails, the other vm will be up and running.

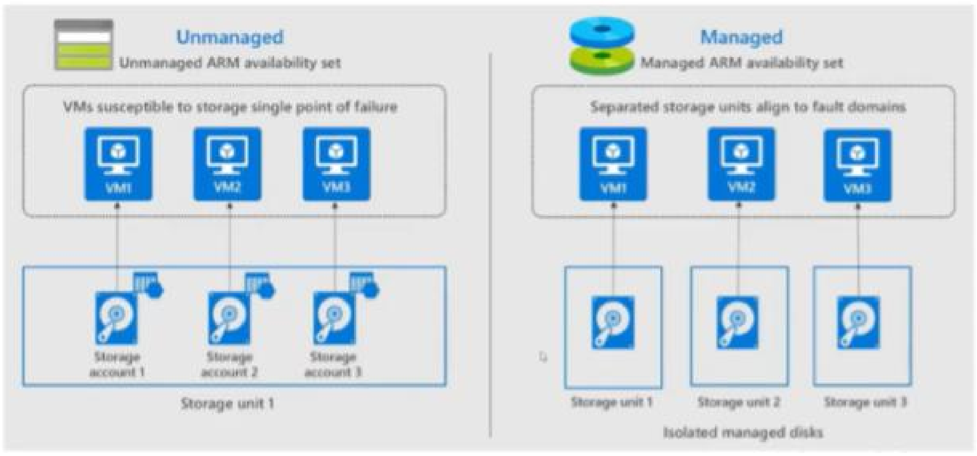


Figure 3.23: Availability Set
[52]

- **Integration with Availability Zones**:Managed disks support Availability Zones(See section 3.2)
- **Azure Backup support**: To handle regional disaster, Azure provides backup service for the vm or the managed disk which support disk sizes up to 32 TB.
- **Granular access control**: A managed disk is always experienced to(read ,write and update) operations also it can be created deleted by a user in the operation team which is given permissions to work on those disks using (RBAC) ,a user can be assigned the export action to copy the managed disk to a storage account.

- **Upload your vhd**: With this feature you can upload your data directly to managed disk without attaching them to a vm ,you can also upload on –premises virtual machine to Azure in turn it will be uploaded to large managed disks with the ability to be simply backed up and restored .uploading 32 TB vhd is supported.

- **Encryption**:Managed disks support two types of encryption. One is performed by the storage service and is referred to as server side Encryption (SSE), the other type is applied through OS on data Disks for the VMs, this type is called Azure Disk Encryption (ADE).

- **Server-side encryption**:Azure Server-side Encryption offers encryption-at-rest and protects your data to fulfill your organizational security and compliance restrictions. By default (SSE) is applied to all managed disks, snapshots, and images, in all the regions but (Temporary disks, on the other hand, are not encrypted by Storage Service Encryption. You can let Azure to manage your keys through platform-managed keys, keys management will be your responsibility which is known as customer-managed keys.

- **Azure Disk Encryption**:With Azure Disk Encryption you are able to encrypt the OS and Data disks used by an IaaS Virtual Machine.. For Windows, the drives are encrypted using industry-standard BitLocker encryption technology. For Linux, the disks are encrypted using the DM-Crypt technology. The encryption process is integrated with Azure Key Vault to allow you to control and manage the disk encryption keys

- **Disk role**: There are three main disk roles in Azure: the data disk, the OS disk, and the temporary disk. These roles relate to disks that are attached to your virtual machine.

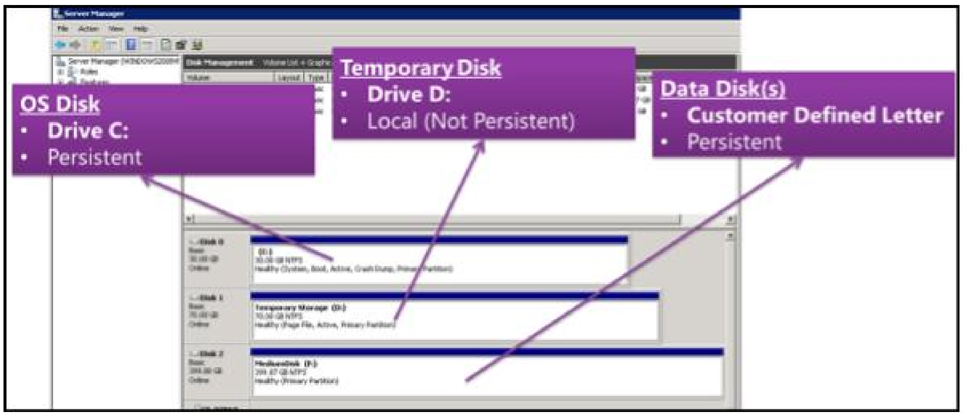


Figure 3.24: VM Disk
[53]

- **Data disk**:A data disk is a managed disk that's associated with a virtual machine to store application data, or other data you want to preserve. Data disks are registered as SCSI drives and are labeled with a letter that you choose. Each data disk has a maximum capacity of 32,767 (GB). The size of the virtual machine determines how many data disks you can attach to it and the type of storage you can use to host the disks.

- **OS disk**: A VM has one attached operating system disk. That OS disk has a pre-installed OS, which was selected when the VM was created. This disk contains the boot volume. This disk has a maximum capacity of 2,048 GB.

- **Temporary disk**: A temporary disk is available for each VM and is not a managed disk. The temporary disk function is to offer short-term storage for applications and processes and is associated to only save data such as page or swap files. Data on the temporary disk may be removed during a maintenance event or when you recreate a VM.but keep in mind during a successful standard reboot of the VM, the data on the temporary disk will be kept. On Azure Linux VMs, the temporary disk is literally /dev/sdb and on Windows VMs the temporary disk is D: by default. The temporary disk is not encrypted by Server Side Encryption.

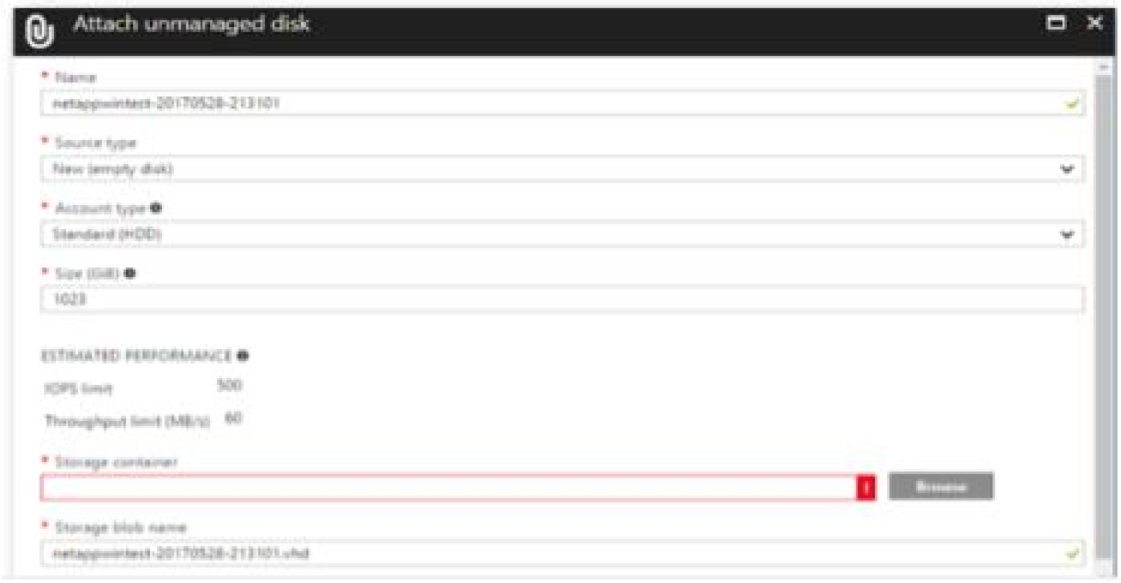
- **Managed disk snapshots** A managed disk snapshot is a read-only full copy of a managed disk in a specific time that is saved as a standard managed disk by default. With snapshots, you can take a backup of your managed disks at any point in time. These snapshots are independent of the source disk and can be used to create new managed disks. Snapshots are billed based on the used size. For example, if you create a snapshot of a managed disk with provisioned capacity of 64 GiB and actual used data size of 10 GiB, that snapshot is billed only for the used data size of 10 GiB. An Azure usage Report of your snapshots can detail the size of all snapshots. For example, if the used data size of a snapshot is 10 GiB, the daily usage report will show 10 GiB/(31 days) = 0.3226 as the consumed quantity.

- **Images**:Managed disk can provide deploying a managed custom image .you have two ways to create an image either from a custom VHD into a storage account or through using (sysprepped) VM. In turn this image will include all the attached disks of the VM containing both the OS and Data Disks, this image can be used for deploying several of VMs using your custom image without requiring any storage account.

- **Images versus snapshots**:How I can tell the difference between images and snapshots. With managed disks, you can take an image of a generalized VM that has been de-allocated. This image contains all the associated disks to the VM.in turn this image can be utilized to create a VM, and it includes all of the disks. A snapshot is like taking a copy of a disk at a specific point of time. It is related to one disk. If you have a VM that has one disk (the OS disk), you can take a snapshot or an image of it and create a VM from either the snapshot or the image. A snapshot doesn't recognize any disk except the one it has. It will be useless in scenarios where a VM contains multiple disks like striping.

3.7.15.1 Unmanaged disks

Is a (.vhd) file stored on a Azure Storage account, not an Azure Resource Manager Whereas the managed disk is “Azure Resource Manager”


Attach unmanaged disk

* Name
netappointest-20170528-213101

* Source type
New (empty disk)

* Account type
Standard (HDD)

* Size (GiB)
1023

ESTIMATED PERFORMANCE

IOPS limit 500

Throughput limit (MB/s) 60

* Storage container

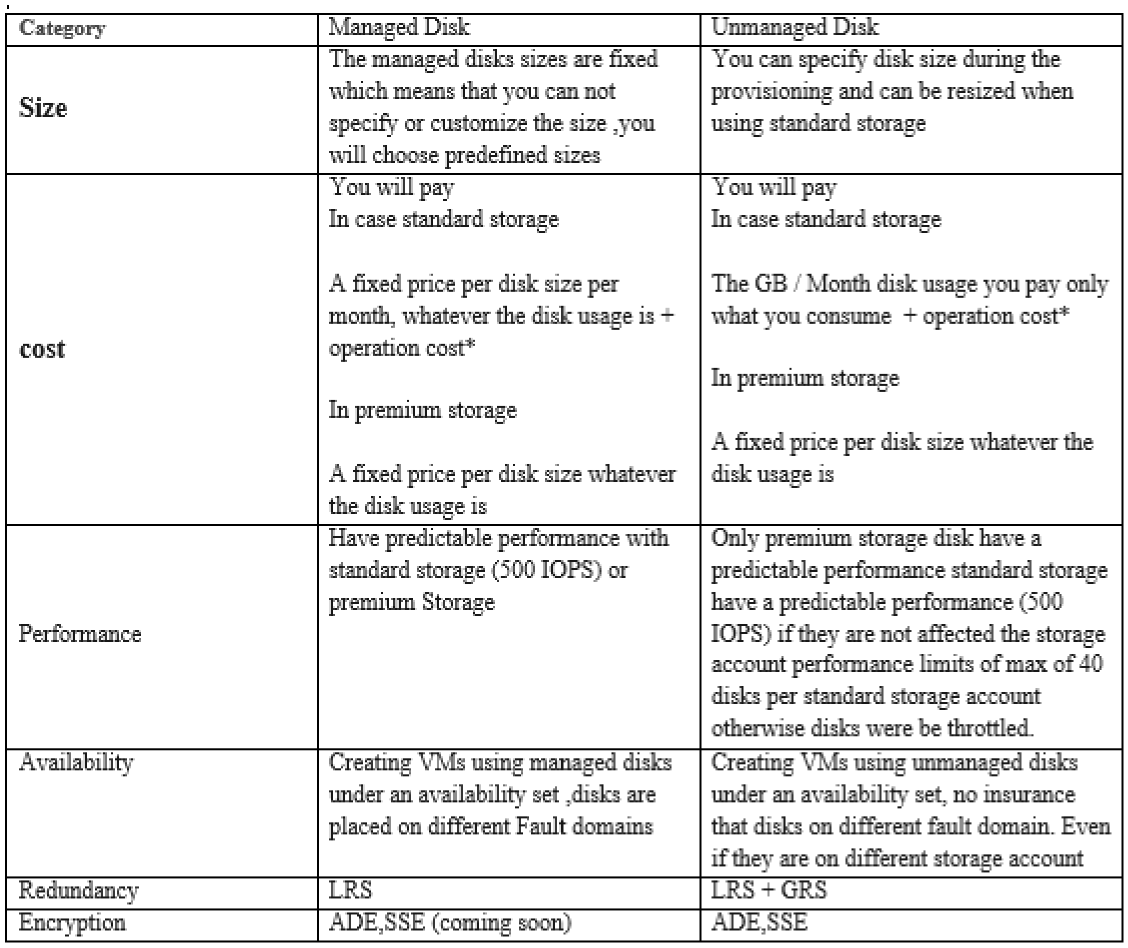
* Storage blob name
netappointest-20170528-213101.vhd

Figure 3.25: Create Unmanaged Disk
[53]

As shown above you can define the disk size while creating the disk (and can be resized) when using Standard Storage.(un-managed disk) [54]

3.7.15.2 Comparison between Managed and Unmanaged Disk

To calculate the cost we have to know the following item.



Category	Managed Disk	Unmanaged Disk
Size	The managed disks sizes are fixed which means that you can not specify or customize the size ,you will choose predefined sizes	You can specify disk size during the provisioning and can be resized when using standard storage
cost	You will pay In case standard storage A fixed price per disk size per month, whatever the disk usage is + operation cost* In premium storage A fixed price per disk size whatever the disk usage is	You will pay In case standard storage The GB / Month disk usage you pay only what you consume + operation cost* In premium storage A fixed price per disk size whatever the disk usage is
Performance	Have predictable performance with standard storage (500 IOPS) or premium Storage	Only premium storage disk have a predictable performance standard storage have a predictable performance (500 IOPS) if they are not affected the storage account performance limits of max of 40 disks per standard storage account otherwise disks were be throttled.
Availability	Creating VMs using managed disks under an availability set ,disks are placed on different Fault domains	Creating VMs using unmanaged disks under an availability set, no insurance that disks on different fault domain. Even if they are on different storage account
Redundancy	LRS	LRS + GRS
Encryption	ADE,SSE (coming soon)	ADE,SSE

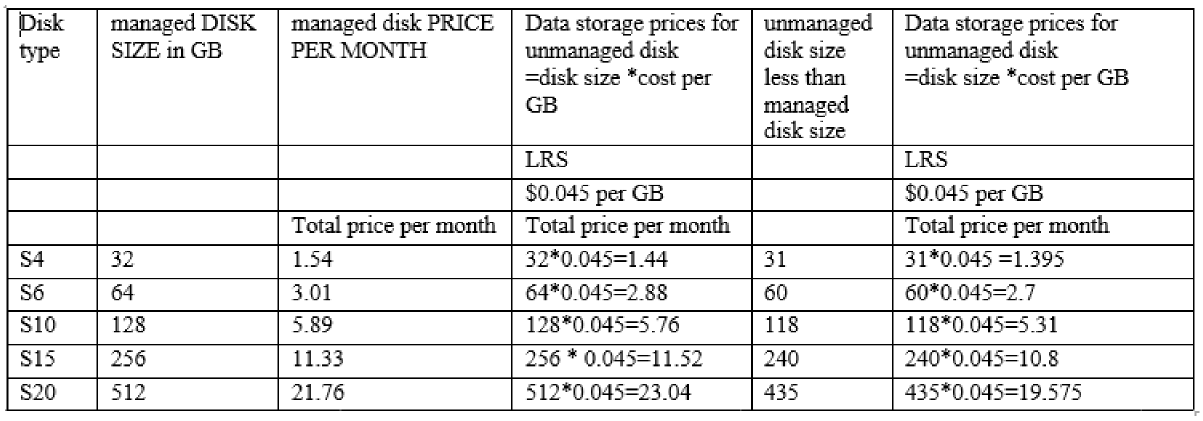
Figure 3.26: Managed Disk versus Unmanaged Disk
[55]

3.7.15.3 Cost in Managed Disk and in Unmanaged disk

- **Operations cost refers to**: Replication data transfer cost (In case of GRS) + Storage operations costs.
- **Managed disks**are more expensive than un managed disks but keep in mind that It depends, in some scenarios, un managed disks cost more or equal to managed disks.

  Standard Storage managed disk cost per month Managed disk cost = Fixed Cost (Per disk size) + Operations cost Un-managed disk cost = (Storage Usage In GB * Cost per GB + Operations cost)
  Because the Operations cost is fixed for both models, it will be removed during calculation. Because the managed disk pricing model is not per usage, we will calculate the Disk size equity value, to be able to compare with un-managed disk.

  Cost for Standard HDD Managed Disks



Disk type	managed DISK SIZE in GB	managed disk PRICE PER MONTH	Data storage prices for unmanaged disk =disk size *cost per GB	unmanaged disk size less than managed disk size	Data storage prices for unmanaged disk =disk size *cost per GB
			LRS		LRS
			$0.045 per GB		$0.045 per GB
		Total price per month	Total price per month		Total price per month
S4	32	1.54	32*0.045=1.44	31	31*0.045 =1.395
S6	64	3.01	64*0.045=2.88	60	60*0.045=2.7
S10	128	5.89	128*0.045=5.76	118	118*0.045=5.31
S15	256	11.33	256 * 0.045=11.52	240	240*0.045=10.8
S20	512	21.76	512*0.045=23.04	435	435*0.045=19.575

Figure 3.27: Price of managed disk versus un-managed disk
[56]

From the previous table ,If you use unmanaged hard disk sizes less than the managed hard disk size, then unmanaged disks will cost less than managed disks. If you use more or equal to the managed hard disk size, then managed disks cost will be less than unmanaged disks. but the answer is very relative to your needs. As you might know that there are many benefits of using managed disks:

- Disk snapshots
- Predictable performance
- Distribution in different fault domains when associated with Availability Sets.
- ARM object

3.8 Azure Security

The purpose of security is protecting data from unauthorized access in order to accomplish full protection we should determine who can access the data in addition encrypting the data itself. In the later section we will demonstrate the different methods designed by Azure for controlling access to data and assigning different permission to manipulate stored data.

3.8.1 Azure Techniques to Control access to account data

The storage account owner is the only one who can access data and the only one who can delegate to other users for accessing data and assign them the appropriate permission. Authorization has to take place at each request directed to the storage account, therefore any request must contain the Authorization header which has all the required information for validating the service then running it. Assigning access to the data in storage account using any of the following techniques:

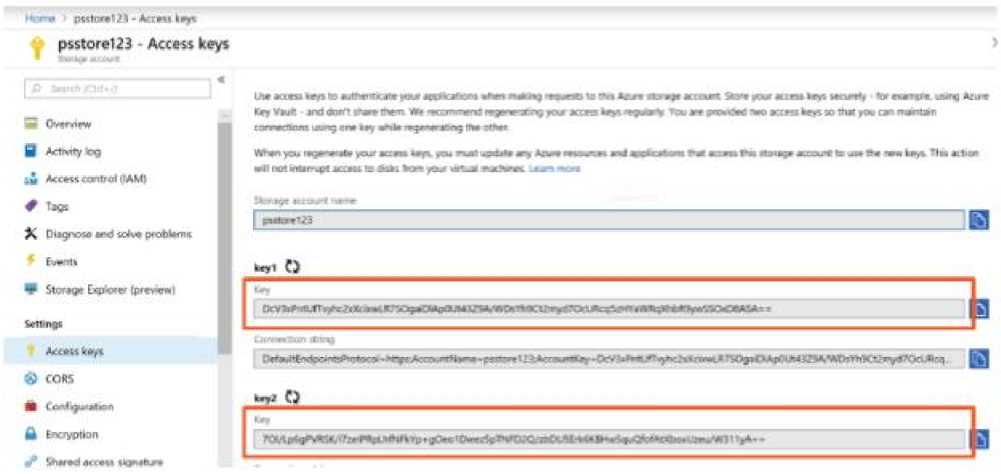
1. Azure Active Directory
2. Shared Key authorization
3. Shared access signature
4. Container Permissions
5. Encryption for Data at Rest
6. Firewalls and Virtual Networks

3.8.2 Azure Active Directory:

Authentication can be provided by Azure Active Directory (Azure AD),when a successful authentication made by an identity ,the Azure AD responds with a token to be used on authorizing the request to Blob storage or queue storage.

3.8.3 Shared Key authorization:

Navigate to your storage account access key to build a connection string that your application needs it to access Azure Storage during running. The values in the connection string is used to build the Authorization header that is passed to Azure Storage. If this key is given to any user it will have a full control over the storage account ,the following figure shows access key therefore shared access signature is more granular .


Home > psstore123 - Access keys

psstore123 - Access keys
Storage account

Search (Ctrl+/)
- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events
- Storage Explorer (preview)

Settings
- Access keys
- CORS
- Configuration
- Encryption
- Shared access signature

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your access keys securely - for example, using Azure Key Vault - and don't share them. We recommend regenerating your access keys regularly. You are provided two access keys so that you can maintain connections using one key while regenerating the other.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action will not interrupt access to disks from your virtual machines. Learn more

Storage account name
psstore123

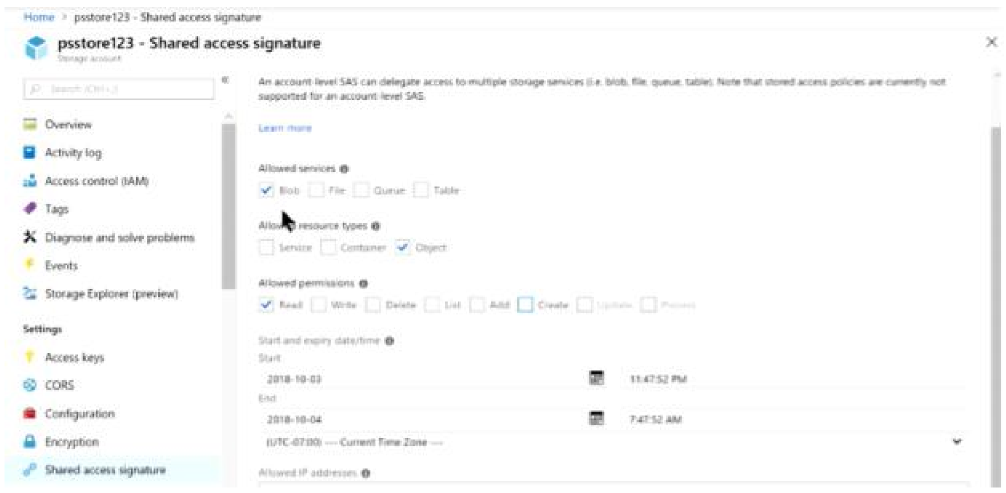
key1
Key
DcV3xPrtUfTvyhc2xXciswLR7SOgaDlAp0Ut43Z9A/WDsYh9Ct2myd7OcURcq5zHYaWRqXhbR9yw5SOxD8ASA==
Connection string
DefaultEndpointsProtocol=https;AccountName=psstore123;AccountKey=DcV3xPrtUfTvyhc2xXciswLR7SOgaDlAp0Ut43Z9A/WDsYh9Ct2myd7OcURcq...

key2
Key
7OI/LpfqPVRSK/l7zeIPRpLhfNfkYp+gOeo1Dwez5pTNFD2Q/zbDUSErk6KBHw5quQfofAtXboxUzeu/W311yA==

Figure 3.28: Access Key
[56]

3.8.4 Shared access signature: [1]

Use a shared access signature to delegate access to resources in your storage account, here you can select what services are allowed to the user from the following services (Blob, File,queue,Table). A shared access signature is a token that encapsulates all of the information needed to authorize a request to Azure Storage on the URL. You can identify the storage resource, the permissions granted, and the interval over which the permissions are valid as part of the shared access signature. The following Figure declare the shared access signature.


Home > psstore123 - Shared access signature

psstore123 - Shared access signature
Storage account

Search (Ctrl+/)
- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events
- Storage Explorer (preview)

Settings
- Access keys
- CORS
- Configuration
- Encryption
- Shared access signature

An account-level SAS can delegate access to multiple storage services (i.e. blob, file, queue, table). Note that stored access policies are currently not supported for an account-level SAS.

Learn more

Allowed services
[✓ Blob] File Queue Table

Allowed resource types
Service Container [✓ Object]

Allowed permissions
[✓ Read] Write Delete List Add Create Update Process

Start and expiry date/time
Start
2018-10-03 11:47:52 PM
End
2018-10-04 7:47:52 AM
(UTC-07:00) — Current Time Zone —

Allowed IP addresses

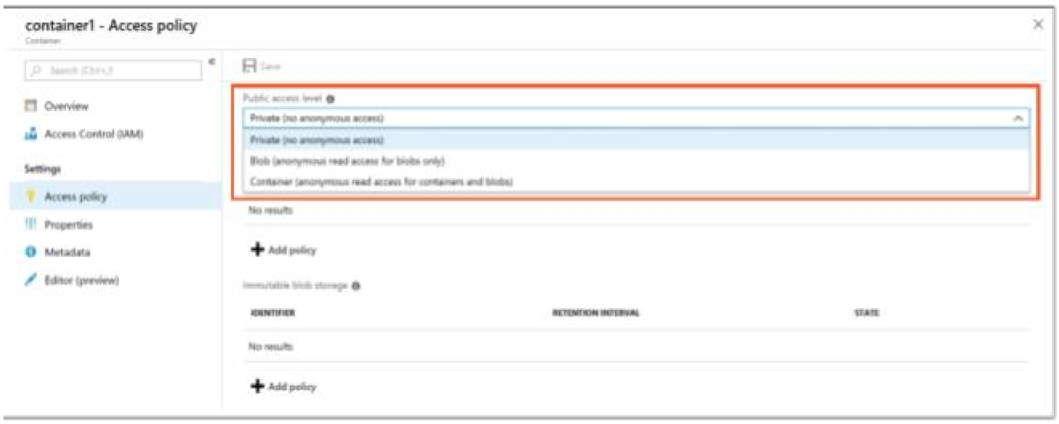
Figure 3.29: Shared Access Signature
[56]

3.8.5 Container Permissions

There is a set of configuration options that can be applied to the container level,which means that all of the blobs in that container will inherit the same permission. If you need to get granular about your blob security configuration, you can create multiple containers so you can configure them separately.

By default “private no anonymous access “is permitted which means that only authenticated users can access the blob .

If you need to open the blob up for public access, then you have two choices, Blob or Container. Blob refers to the anonymous availability for each blob in that container without any authentication. If the blob is an image, then typing the image’s URL into the web browser will actually show the image. The previous setting is Container, which also opens all blobs up for public access just like the blob setting, but it also opens the container up for public access.


container1 - Access policy
Container

Public access level
Private (no anonymous access)
Private (no anonymous access)
Blob (anonymous read access for blobs only)
Container (anonymous read access for containers and blobs)

No results

Immutable blob storage

IDENTIFIER	RETENTION INTERVAL	STATE
No results		

Figure 3.30: Access policy
[56]

3.8.6 Encryption for Data at Rest [2]

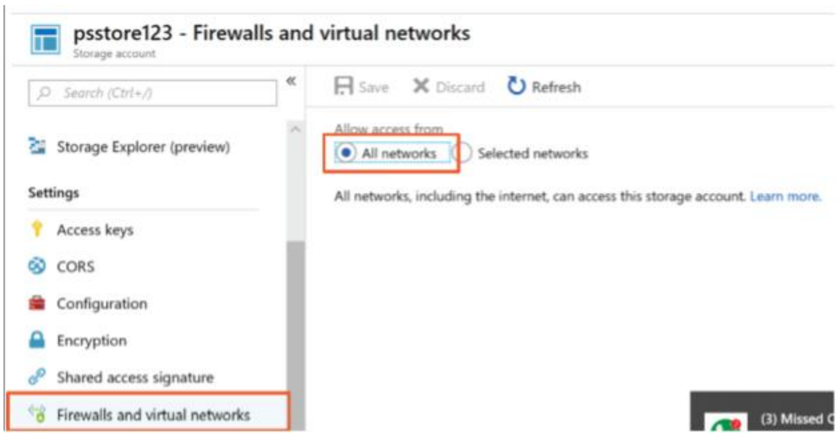
You need to guarantee that data at rest must be encrypted so no one can view data excep the one who owns the encryption key .
Encryption at Rest is the encryption of data when it is stored. The Encryption at Rest designs in Azure use symmetric encryption to encrypt and decrypt big amounts of data quickly:

* A symmetric encryption key is used to encrypt data as it is written to storage.
* The same encryption key is used to decrypt that data as it is queried for use in memory.
* Data may be partitioned, and different keys may be used for each partition.
* Keys must be stored in a secure location with identity-based access control and audit policies. Data

encryption keys are often encrypted with a key
encryption key in Azure Key Vault to further limit access.

3.8.7 Firewalls and Virtual Networks

The firewalls and virtual networks feature of Azure Storage permits you to strictly access your storage account for certain networks or IP address ranges, one of the most important benefits of the internet and the cloud is that global accessibility. That accessibility is kind of a threat therefore when you identify who you want to have access to your data. Every object in your storage account has a URI endpoint, and requests can be made to it from anywhere in the world. They might come from known IP address ranges, like maybe your office building or that of your partners. They might come from another application or a virtual machine in Azure that's part of a virtual network, or they may come from anywhere else. With firewalls, we can allow certain IP address ranges, and with virtual networks, we can allow certain Azure virtual networks.as shown in Figure below.



Allow access from
[✓ All networks] Selected networks

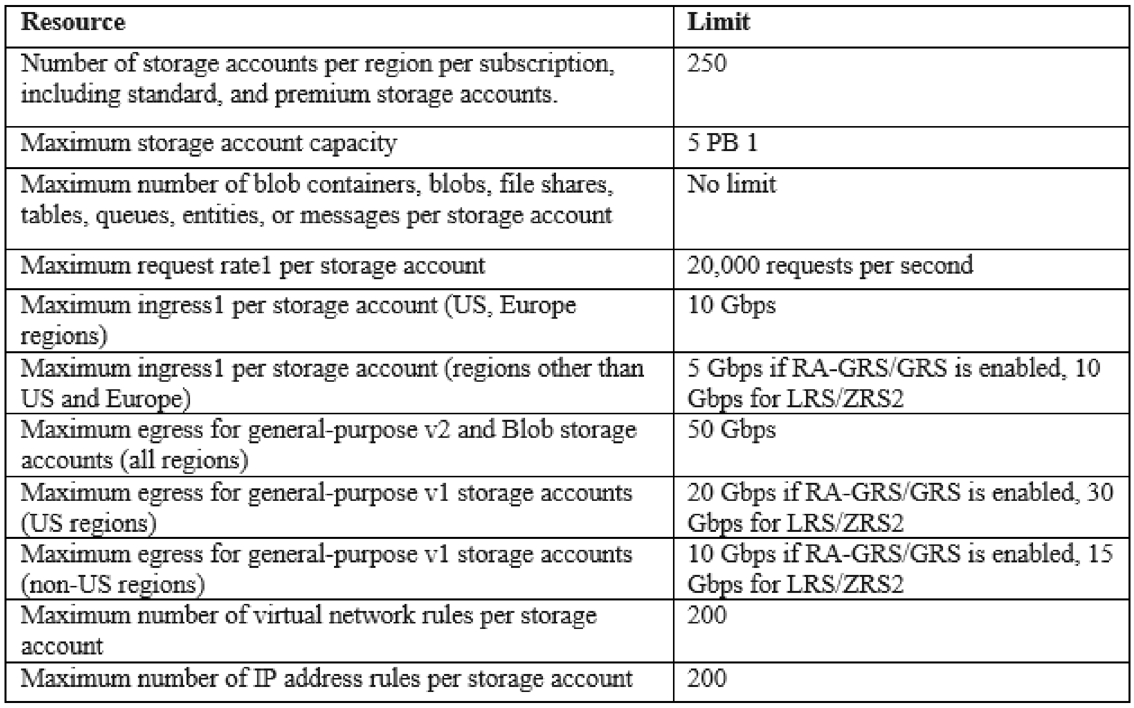
All networks, including the internet, can access this storage account. Learn more.

Figure 3.31: Firewall and Virtual Network.
[57]

3.8.8 Azure storage account scale targets

A storage account is a pool of storage in Azure that can be used by multiple storage services, including Azure Files, to store data. Other services that save data in storage accounts are Azure Blob storage, Azure Queue storage, and Azure Table storage.

The following table defines default limits for Azure general-purpose v1, v2, Blob storage, and block blob storage accounts. The ingress limit denotes all data that is sent to a storage account. The egress limit states that all data that is received from a storage account. If the requirements of your application exceed the scalability targets of a single storage account, you can build your application to use multiple storage accounts. You can divide your data objects across those storage accounts.



Resource	Limit
Number of storage accounts per region per subscription, including standard, and premium storage accounts.	250
Maximum storage account capacity	5 PB 1
Maximum number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
Maximum request rate1 per storage account	20,000 requests per second
Maximum ingress1 per storage account (US, Europe regions)	10 Gbps
Maximum ingress1 per storage account (regions other than US and Europe)	5 Gbps if RA-GRS/GRS is enabled, 10 Gbps for LRS/ZRS2
Maximum egress for general-purpose v2 and Blob storage accounts (all regions)	50 Gbps
Maximum egress for general-purpose v1 storage accounts (US regions)	20 Gbps if RA-GRS/GRS is enabled, 30 Gbps for LRS/ZRS2
Maximum egress for general-purpose v1 storage accounts (non-US regions)	10 Gbps if RA-GRS/GRS is enabled, 15 Gbps for LRS/ZRS2
Maximum number of virtual network rules per storage account	200
Maximum number of IP address rules per storage account	200

Figure 3.32: Scalability Targets
[58]

4 Decentralized storage network

The notion of a Decentralized Storage Network (DSN) is a self-coordinate network between untrusted parties to aggregate storage offered by multiple independent storage providers to provide data storage and data retrieval to clients. DSNs have been gaining an increased mainstream adoption recently due to the phenomenal hype of bitcoin's and Ethereum's success that highlights that blockchain is more than just a niche technology for the financial sector, It can offer a new, decentralized, trusted, and transparent platform that could benefit a wider range of industries(e.g., financial, medical, public utilities, identity management, asset registration, government agencies, etc.). This interest in exploring all the potential successful blockchain use cases and making the most of blockchain's benefits is not far away from the storage domain as data is the most valuable asset for any business in all business sizes. Also, many people have some free storage space on their computer drive and it could be useful to rent it out creating a secure and incentive model by embedding blockchain technology.

This section is divided into three main parts, in the first part, we briefly give a primer on p2p, and highlight some P2P networks for file exchange that have significantly impacted the p2p development. In the second part, we are introducing the blockchain technology and highlighting two main successful blockchain projects. In the third part, we are introducing some decentralized storage networks and digging deeper in IPFS and Filecoin as two of the major players in this domain.

4.1 Peer to peer file exchange networks

4.1.1 Peer to peer (P2P) Overview:

The term P2P [59]refers to an application-level overlay network where every peer/node "machine" acts as client and server at the same time. P2P network's Peers are equally privileged, equipotent participants whereas the client-server model is a privilege based; dedicated servers for specific clients.

4.1.2 Characteristics:

A P2P network should be symmetric; no special capability of certain nodes, decentralized by nature, operate with un managed volunteer participants, secure from attacks and system failure, and scalable; the capability to serve millions of peers efficiently while this is hard to be achieved with the client-server model that can serve up to thousands [60].

4.1.3 History:

P2P is not a new network topology, the internet in its early beginning was P2P and over time it has become increasingly a client-server network topology. In 1969, Stanford Research Institute, UC Santa Barbara, and the University of Utah have created the ARPANET,which is considered the basis of the current internet. The main goal of ARPANET was to share computing resources around the U.S.. [61]

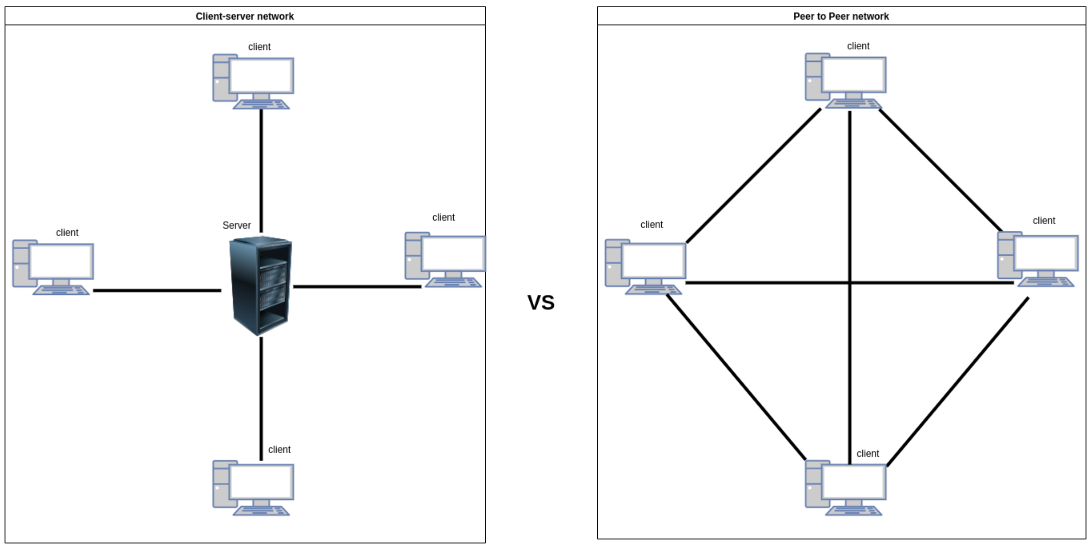


Figure 4.1: Client-server network vs P2P network figure
[59]

4.1.4 Examples of Peer to Peer Network

4.1.4.1 Napster

Napster is a pioneering P2P multimedia file sharing application founded in 1999 and developed by Shawn Fanning while still a freshman at Northeastern University. Napster was developed as a free program that users can download to search for MP3 files in the local disks of neighboring computers. Users can download these files directly from each other. Napster gained significantly increased adoption from users; in less than one year, Napster had over one million members. Napster protocol was designed to let users swap files to each other but in order to answer search queries and mediate client connections, a centralized Napster index server was implemented. Napster's increased adoption inflamed the copyright war because sharing the MP3 files even before being released was hurting the sales record industry so badly. On December 6, 1999, the Recording Industry Association of America raised the first lawsuit against Napster, and on July 11, 2001, Napster shut down its entire network to comply with the injunction. Although Napster's peers were equally privileged and it was powered by its users, shutting down Napster's index servers was enough to make these peers unable to communicate. [62]

4.1.4.2 Gnutella

Gnutella is the first fully decentralized peer-to-peer network of its kind mainly used for file searching and sharing [63]. It was founded In early 2000 and developed by Justin Frankel and Tom Pepper of Nullsoft but currently, it's led by the Gnutella Developers Forum [64]. To join the system, a new Gnutella node called servant which has a Gnutella client software, initially must bootstrap and find at least one other node. In most cases, a servant might connect to one of several known hosts that are almost always available (e.g., gnutellahosts.com). Once connected, the client software requests a list of working addresses and tries to connect to until it reaches a certain quota. Nodes send messages to each other to interact with each other. Messages could be broadcasted or back-propagated. The messages allowed in the network are Group Membership (PING and PONG) Messages to check the node connectivity status, Search (QUERY and QUERY RESPONSE) Messages, and File Transfer (GET and PUSH) messages. [65] [66]

4.2 Blockchain

Blockchain as a technology, at its heart, is a self-sustaining, peer-to-peer electronic ledger system for managing and recording any type of transactions with no central authority involvement. Blockchain is cryptographically secure by design as each "block" is uniquely connected to its previous blocks by including the previous block hash " called parent block" in the new block " called child block". Verification is handled through consensus algorithms among multiple nodes making blockchain platforms immutable, and updatable only via consensus, an agreement among nodes. Thus remove the need for any trusted third parties and make blockchain platforms protected against the domination of the network by any single node or group of nodes raising the new concept of "nodes' democracy ". Each block basically contains a block header, contains the parent hash, the Merkle root of the tree-like transaction information, and a timestamp, and block body which contains a series of transactions.

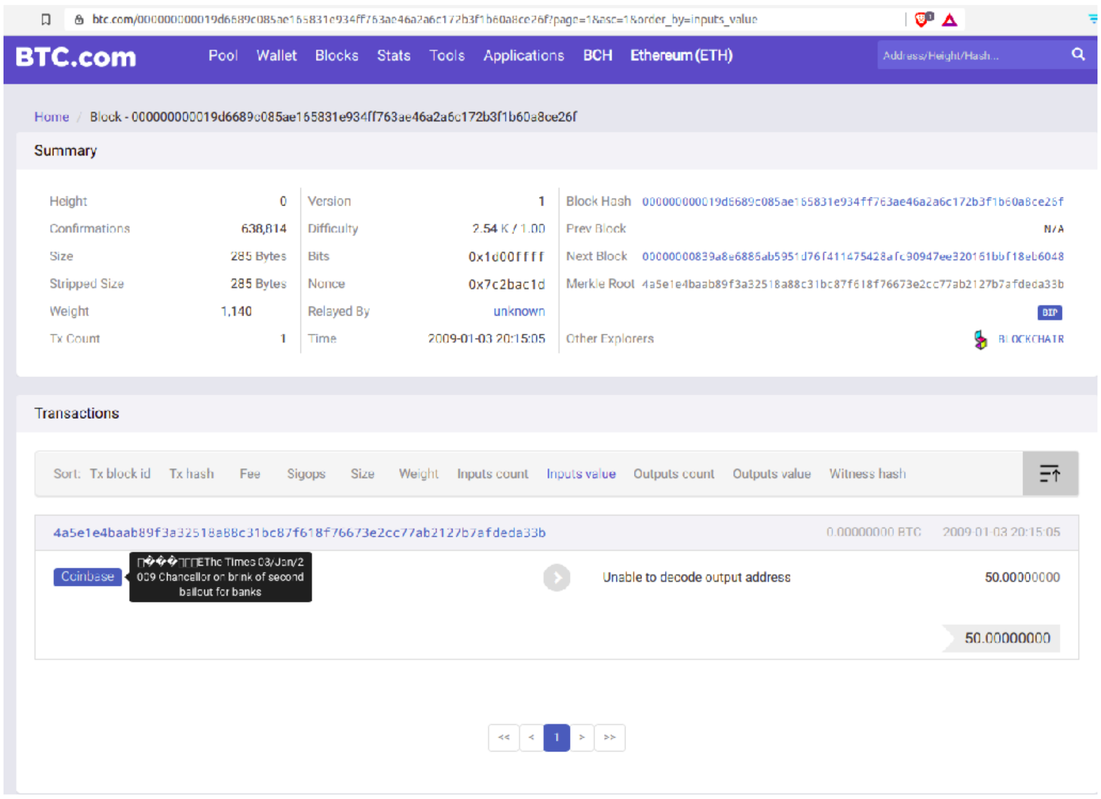
4.2.1 Blockchain networks topology

Most of the blockchain networks come in three main types:

* **Permissionless public blockchain network:** an open network where any node can join, participate in consensus mechanism, read, write, and validate the data. Bitcoin, Ethereum, Ripple [67], and Zcash [68]are examples of public blockchain networks.
* **Permissioned private blockchain network:** a closed restrictive environment that works under the control of an entity that reserves the right to provide access to the network. Also, read-and-write access to the entities has to be provided on demand. Hyperledger [69] is an example of permissioned blockchain networks.
* **Consortium blockchain network:** A hybrid type of the private and public networks controlled by preselected nodes but open so that any node could join and validate the data. MultiChain [70], Quorum [71], HyperLedger, Tendermint [72], etc are examples of consortium networks.

4.2.2 Bitcoin

The most prominent example of a successful blockchain deployment is the Bitcoin blockchain network, A Peer-to-Peer Electronic Cash System, it's an open, borderless, neutral censorship-resistant public blockchain network with a built-in digital currency called bitcoin. In October 2008 bitcoin's whitepaper authored by Satoshi Nakamoto[pseudonymous] was sent to a mail-list and in January 2009, the bitcoin network came into existence with Nakamoto mining the genesis block [73] " the first block, number 0" with a hidden message "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks" inside the coin base transaction [74], as shown in the figure below.


btc.com/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f?page=1&asc=1&order_by=inputs_value
BTC.com
Pool
Wallet
Blocks
Stats
Tools
Applications
BCH
Ethereum (ETH)
Address/Height/Hash...
Home / Block - 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Summary

Height	0	Version	1	Block Hash	000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
Confirmations	638,814	Difficulty	2.54 K / 1.00	Prev Block	N/A
Size	285 Bytes	Bits	0x1d00ffff	Next Block	00000000839a8e6886ab5951d76f411475428afc90947ee320161bbf18eb6048
Stripped Size	285 Bytes	Nonce	0x7c2bac1d	Merkle Root	4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b
Weight	1,140	Relayed By	unknown		BIP
Tx Count	1	Time	2009-01-03 20:15:05	Other Explorers	BLOCKCHAIR

Transactions

Sort: Tx block id Tx hash Fee Sigops Size Weight Inputs count Inputs value Outputs count Outputs value Witness hash

4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b 0.00000000 BTC 2009-01-03 20:15:05

Coinbase ��� EThe Times 03/Jan/2009 Chancellor on brink of second bailout for banks

Unable to decode output address 50.00000000

50.00000000

Figure 4.2: Bitcoin block 0
[73]

4.2.2.1 Bitcoin Novelty

The novelty of bitcoin is in proposing a solution for double-spending without trusting any third parties, proposing public transactions with high privacy coverage, and creating an incentive model for participants to maintain the network secured. Timestamp server which works by taking a hash of a block of items to be timestamped and widely publishing the hash is used to prevent double-spending. Bitcoin uses Proof of Work (POW) [75] as a consensus algorithm which is a computational puzzle, hard to solve but easy to check and verify for majority decision making and ensures network resistance against attacks (e.g. Sybil attack) [76]. If a malicious node tries to attack the network by modifying a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. Mining which is the action of creating a new block and getting block reward as well as collecting transaction fees, tips for miners to incentive them to include your transactions before others, via finding the POW, is used as an incentives model for nodes to maintain the network secure. The protocol also halves the rate at which new bitcoins are created as mining rewards every 4 years and limits the total number of bitcoins that will be created to a fixed total of 21 million coins by the year 2140 [77]. Digital signatures [78] and Public-key cryptography [79] in sending any transaction are core components to provide privacy and authentication. Thus provide privacy coverage as the public can see that someone "string of random numbers " is sending a transaction to someone else "string of random numbers ", but without information linking the transaction to anyone. [80]

4.2.2.2 Bitcoin life cycle

The bitcoin lifecycle could be summarized in 5 steps:

1. New transactions are broadcast to all nodes
2. All nodes collect these broadcasted transactions and validate them.
3. Miner nodes compete to find the POW.
4. When any miner finds the POW, immediately broadcasts the new block to all nodes.
5. All nodes accept the new block as long as the transaction is valid start working on the new block.

Bitcoin block as illustrated in figure 4.3 is made of a header, containing block metadata, followed by a long list of transactions. The block header is 80 bytes, whereas the average transaction is at least 250 bytes and the average block contains more than 500 transactions. A complete block, with all transactions, is therefore 1,000 times larger than the block header. Block header consists of three sets of block metadata. First, there is a reference to a previous block hash, which connects this block to the previous block in the blockchain. The second set of metadata, namely the difficulty, timestamp, and nonce, relates to the mining competition. The third piece of metadata is the Merkle tree root, a data structure used to efficiently summarize all the transactions in the block as figure number 4.3. The block generation is auto-adjusted by software every 10 minutes.

4.2.3 Ethereum

Ethereum is an open-source, blockchain-based, Turing complete decentralized software platform featuring smart contract functionality with a built-in cryptocurrency, Ether. Ether is generated as a reward to miners for computations performed to create new blocks thus securing the network. Ethereum idea was proposed in late 2013 by Vitalik Buterin [81], a cryptocurrency

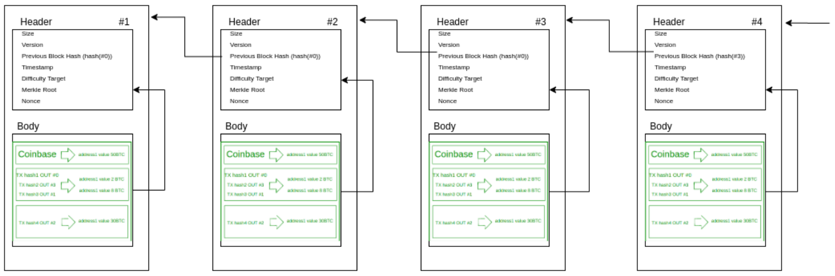


Figure 4.3: Bitcoin block 0
[73]

researcher and programmer, presented at a Bitcoin conference held in Miami between 25th-26th January 2014 and launched in 2015. Development was funded by an online crowdsale that took place between July and August 2014 [82]. The system then went live on 30 July 2015, with 72 million coins minted [83]. Ethereum's goal is to act as the " world Computer" to enable Smart Contracts and Distributed Applications (ÐApps) to be built and run without any downtime, fraud, control, or interference from a third party and it serves as the platform for over 260,000 different cryptocurrencies, including 47 of the top 100 cryptocurrencies by market cap. The network achieves consensus among nodes through POW but Proof of Stack (POS) [84] ,a class of consensus algorithms in which validators vote on the next block, and the weight of the vote depend upon the size of its stake, will be implemented in Eth 2.0 [85]upgrade which aims to increase the transaction throughput sharing technology. [86]

4.2.3.1 Ethereum main building blocks:

- **Smart contract:**A self-executed and self-verified computer code written in rich programming languages(e.g. Solidity [83] Vyper [84]) and compiled into EVM bytecode [85]. Smart contracts run exactly as programmed without any possibility of censorship, downtime, fraud, or third-party interference. Smart contracts allow developers to create whatever application they want on top of the Ethereum platform.
- **Ethereum Virtual Machine (EVM):** EVM [87] [**?**] is a Turing complete stack-based virtual machine. Every Ethereum's node runs an implementation of EVM.
- **Block**:Unlike bitcoin, Ethereum block structure is more complex. Ethereum does not have one Merkel tree but 4 tries/trees as shown in the below figure
  1. **State Trie:** A key-value pair tree for every account in the Ethereum network.
  2. **Storage Trie:** a tree that holds contract data.
  3. **Transaction Trie:**Contains a list of transactions, each block has its own separated trie.

4. **Receipts Trie:** Contains a list of receipts of each transaction in the block.

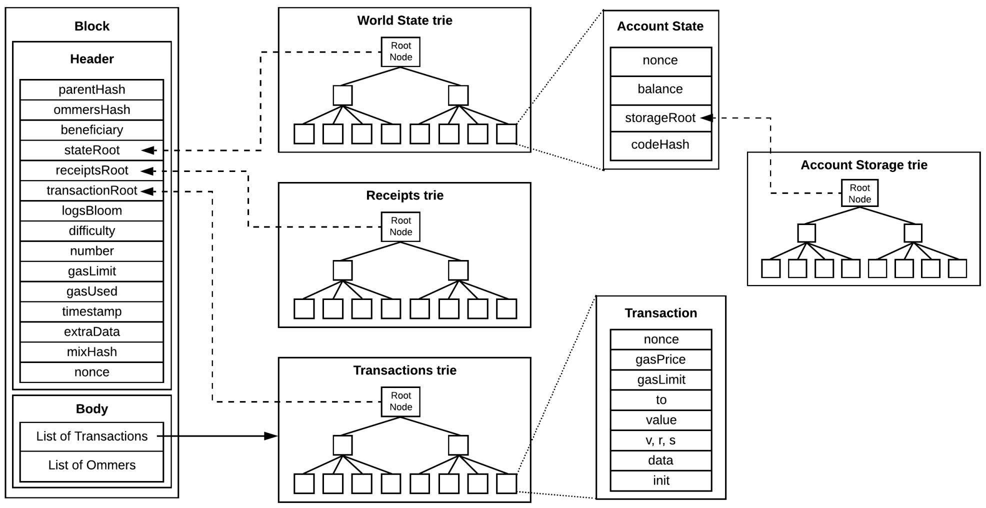


Figure 4.4: Ethereum block
[88]

* **Transactions:** The transaction is the operation of changing the current state to the next state. Ethereum supports two types of transactions:
  - **Message calls:**It could be either transfer value transaction between two EOAs or message call to a contract to execute a specific operation in the contract.
  - **Contract creations:** only for deploying new contract to the network
* **State**: Often called a world state, world state is a mapping between address/account and account states. You can think about it as a global state that is constantly updated by every transaction execution
* **Account types:** Ethereum has two types of accounts :
  1. **Externally Owned Account (EOA):** this type of accounts has a private key and the account owner can execute any transaction and has full control over it.
  2. **Contract Account:** this type of accounts is for contracts and the execution functionalities limited by the contract logic.
* **Ethereum Gas:** The fundamental network cost unit. Paid for exclusively by Ether, which is converted freely to and from Gas as required. Gas does not exist outside of the internal Ethereum computation engine; its price is set by the Transaction and miners are free to ignore Transactions whose Gas price is too low. Gas is used in Ethereum to avoid issues of network abuse and to sidestep the inevitable questions stemming from Turing completeness [89] .

5 Decentralized storage network examples

5.1 Storj

Storj (pronounced "Storage") is a peer-to-peer cloud storage network features an end to end file encryption, file sharding, and a blockchain-based hash table to store and retrieve files. Storj, the open-source network [90], was founded in 2014 by Shawn Wilkenson, the founder of Storj and CEO of Storj Labs. The Storj's ICO (Initial Coin offering) [91] was launched in May 2017 (18 May - 24 May) achieving 100 percent of its target by raising 30,000,000 dollars [92]. Storj network acts as middleware between the users who have free space to be rented (”Storage nodes”) and users who need free space to store files and assets (”Clients”). STORJ tokens are used as a payment method in the Storj blockchain network. [93]
The way Storj works is similar to how Bittorrent works:

1. **Storing data to the network:** A client encrypts by his private key to validate the ownership and provide a higher security level and breaks it up into multiple pieces. The pieces are distributed to peers across the network. When this occurs, metadata is generated that contains information on where to find the data again.
2. **When data is retrieved from the network** the client will first reference the metadata to identify the locations of the previously stored pieces. Storj uses distributed hash tables to locate all the shards and piece them together. Then the pieces will be retrieved and the original data will be reassembled on the client's local machine and the client decrypts the file using his private key.

Storj currently has 19 petabytes (19 million gigabytes) of available capacity and is hosted on thousands of nodes run by individuals and partners based all around the world. There are currently about 3,000 users on the platform. [94]

5.2 Swarm

Swarm is a distributed storage platform and content distribution service, a native base layer service of the Ethereum web3 stack. The objective is to offer peer-to-peer storage and serving solution that is DDOS-resistant, zero-downtime, fault-tolerant and censorship-resistant as well as self-sustaining due to a built-in incentive system which uses peer-to-peer accounting, SWAP " Swarm accounting protocol with a tit-for-tat accounting scheme", and allows trading resources for payment. On the 29th of June 202, the swarm team released the alpha version of Swarm's Bee client [95] [96] which changes network protocol from devp2p [97] to libp2p [98] and announced that the old Swarm client can still be used until the network exists, however no maintenance or upgrades are planned for it and no compatibility. Thus the information mentioned here is based on the new Swarm Bee documentation. Through the API provided by the bzz URL scheme, Swarm supports uploading "POST", appending to existing files and resuming incomplete uploads "PUT". Each uploaded file is chunked into chunks, pieces of data (max 4K), the basic unit of storage and retrieval in the swarm, has a unique Swarm Hash (a.k.a. bzzhash) [99], a cryptographic hash of data constructed using a regular hash function (like SHA3) [100] and designed for the purpose of efficient storage and retrieval in content-addressed storage, both local and networked. It serves as its unique identifier and address, and a created manifest entry which defines a mapping between arbitrary paths and files to represent collections. It also contains metadata associated with the collection and its objects (files). The manifest entry is provided as a response. The upload tag that we can use to monitor the status of the upload will contain the reference to the uploaded file as well as the manifest entry. Downloading is supported by the bzz URL scheme as well via "GET". Uploading and downloading through the network entails costs. [101]

5.3 IPFS(InterPlanetary File System)

The InterPlanetary File System (IPFS) is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. IPFS could be seen as a single BitTorrent swarm [102], exchanging objects within one Git repository [103]. IPFS provides a high throughput content-addressed block storage model [104](a cryptographic hash over the content represents the block or object of content), with content-addressed hyperlinks. IPFS combines a distributed hashtable [105], an incentivized block exchange BitSwap [106], and a Self-certifying namespace. IPFS has no single point of failure, and nodes do not need to trust each other. It was introduced in 2014 by Juan Benet, launched in an alpha version in February 2015, and by October 2015 was described by TechCrunch as "quickly spreading by word of mouth [107]". IPFS has a main implementation written in the Go programming language,go-IPFS [108], but another implementation in JavaScript, js-IPFS [109], thus IPFS nodes could run in client-side and server-side as well. The key differentiation between IPFS and its predecessors is that its predecessors were not designed as infrastructure to be built upon while IPFS end goal is to evolve the web itself and be a novel hypermedia transfer technology that offers global, low-latency, and decentralized distribution. IPFS is in active development and has big community, the current development state is illustrated here [110] [111]

5.3.1 Nodes Types:

There are four available node types in IPFS Companion:

∗ **External**: An instance of an IPFS daemon that runs outside of a web browser process and exposes Gateway and writable API over HTTP at TCP ports.

∗ **Embedded :** js-IPFS instance running in the browser (in-memory).
  · Can't act as an HTTP gateway.
  · Known to be CPU-hungry over time, which may drain your battery.
  · Missing DHT.

∗ **Embedded + chrome sockets** : Embedded js-IPFS to with a true p2p experience in browsers that grant IPFS access to chrome.sockets .

* **Public**: Nodes used as an implicit fallback for its gateway functionality when an external node is offline or an embedded node is used to see list of gateways. [112]

5.3.2 IPFS Stack

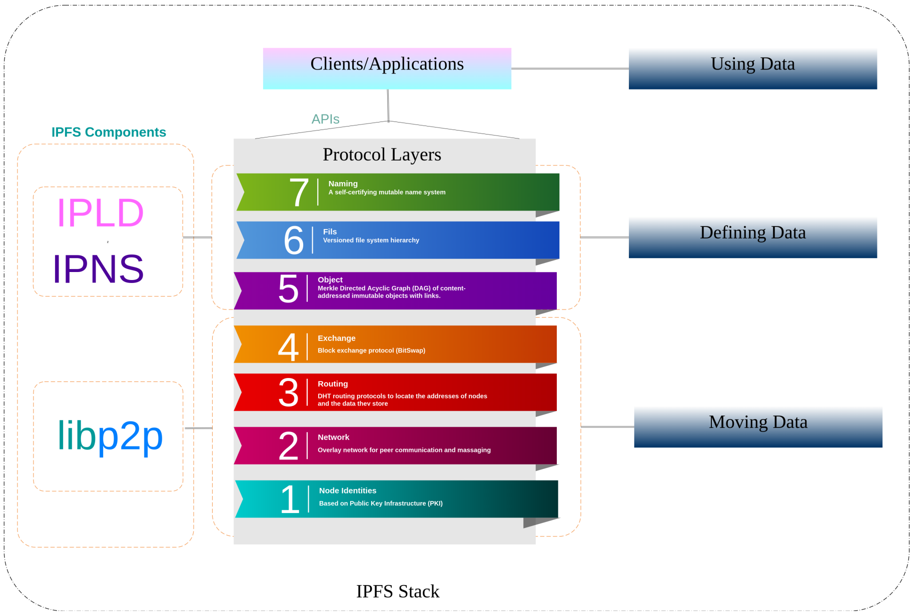

Clients/Applications
Using Data
APIs
IPFS Components
Protocol Layers
IPLD
IPNS
7 Naming
A self-certifying mutable name system
6 Fils
Versioned file system hierarchy
5 Object
Merkle Directed Acyclic Graph (DAG) of content-addressed immutable objects with links.
Defining Data
4 Exchange
Block exchange protocol (BitSwap)
3 Routing
DHT routing protocols to locate the addresses of nodes and the data thev store
libp2p
2 Network
Overlay network for peer communication and massaging
Moving Data
1 Node Identities
Based on Public Key Infrastructure (PKI)
IPFS Stack

Figure 5.1: IPFS Stack

5.3.2.1 IPFS core components

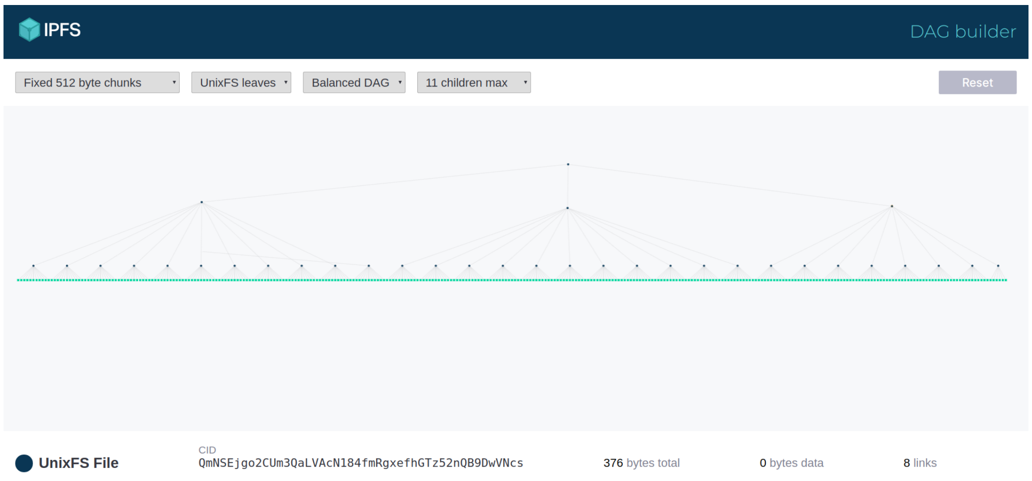
* **IPLD** [113]:The data model of the content-addressable web which combines Merkle trees (used in blockchains to ensure immutability check section 2), and Directed Acyclic Graphs (DAG) [114] (used in Git version control, which also allows users to see the versions of content on IPFS). Any IPLD object consists of 2 components:
  1. Data: A blob of unstructured binary data of size ¡ 256 kB, configurable.
  2. Links: An array of Link structures. Every IPLD Link has 3 parts:
     · Name: Name of the Link
     · Hash: hash of the linked IPFS object, the default hash algorithm is sha256
     · Size: The cumulative size of linked IPFS object, including following its links.

  This adds De-duplication property to IPFS and saves a lot of storage space as duplicated links in any IPDL object are not uploaded.
* **IPNS:** [115] InterPlanetary Naming System, the alternative decentralized version of DNS [116], used to create human-friendly mutable addresses on IPFS which always points to the latest version of the content.

- **Libp2:** A modular network stack to create an overlay p2p network. libp2p is useful for connection multiplexing. Traditionally, every service in a system opens a different connection to communicate with other services of the same kind remotely which is hard to set up and expensive to maintain. Using IPFS, you open just one connection, and you multiplex everything on that. [117]

5.3.2.2 The IPFS Protocol Layer

- **Identities:** This layer manages node identity generation and verification using a cryptographic hash of a public-key, created with S/Kademlia's static crypto puzzle. [118]
- **Network:** The overlay networks for peer communication. IPFS does not require nodes to possess IP addresses and is designed to run on top of any network. to achieve that, IPFS stores addresses as multiaddr [119] formatted bytes string. multiaddr provides away to express addresses and their protocols, including support for encapsulation.
- **Routing:**The routing layer is based on DHTs (Distributed Hash Table) [120]like S/Kademlia that makes a distinction for values stored based on their size. Small values (equal to or less than 1KB) are stored directly on the DHT while in larger values, NodeIds of peers who can serve the block are stored as references. The routing layer's purpose is to find other peers' network addresses, peers who can serve particular objects When data is requested by a node within the network.
- **Exchange:**: IPFS implements BitSwap, BitTorrent inspired protocol. Once the requested data is located, Nodes exchange the data they store using the IPFS BitSwap exchange protocol.
- **Object:**Data structure of data using Merkle DAG as shown in the below figure where links between objects are cryptographic hashes of the targets embedded in the sources. Merkle DAGs provide IPFS many useful properties, including: [121]
  - **Content Addressing:** all content is uniquely identified by its multihash checksum, including links.
  - **Tamper resistance:** all content is verified with its checksum. If data is tampered with or corrupted, IPFS detects it.
  - **Deduplication:** all objects that hold the exact same content are equal and only stored once.



UnixFS File

CID
QmNSEjgo2CUm3QaLVAcN184fmRgxefhGTz52nQB9DwVNcs

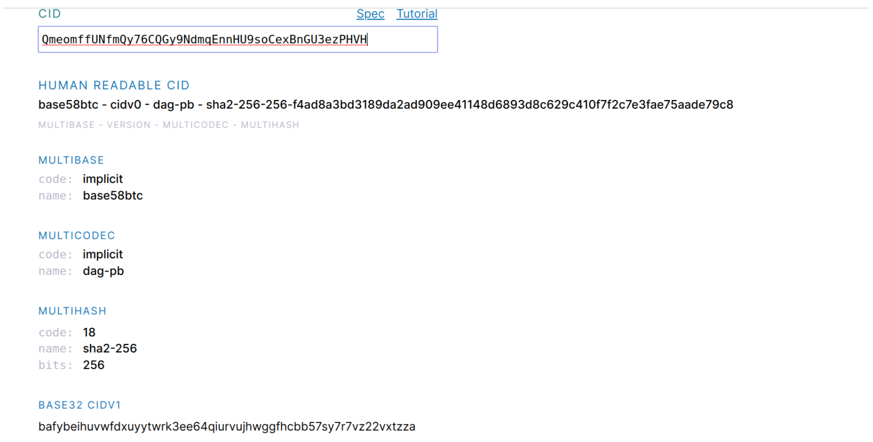
376 bytes total

0 bytes data

8 links

Figure 5.2: DAG Figure
[122]

Each dag node has a unique CID, the hash of the child's dag node. CID is a self-describing content-addressed identifier for distributed systems which doesn't indicate where the content is stored. It uses several multi formats to achieve flexible self-description, namely multi hash for hashes, multi codes for data content types, and multi base to encode the CID itself into strings [123] as shown in the figures below.



CID

QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHVH

HUMAN READABLE CID

base58btc - cidv0 - dag-pb - sha2-256-256-f4ad8a3bd3189da2ad909ee41148d6893d8c629c410f7f2c7e3fae75aade79c8

MULTIBASE - VERSION - MULTICODEC - MULTIHASH

MULTIBASE

code: implicit
name: base58btc

MULTICODEC

code: implicit
name: dag-pb

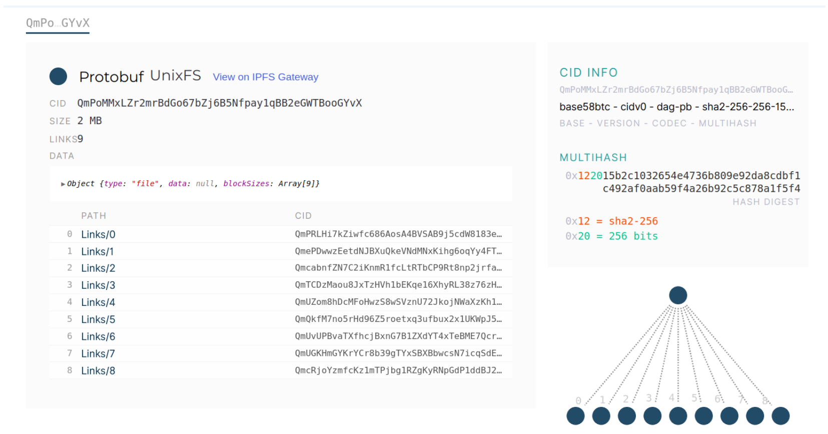
MULTIHASH

code: 18
name: sha2-256
bits: 256

BASE32 CIDV1

bafybeihuvwfdxuyytwrk3ee64qiurvujhwggfhcbb57sy7r7vz22vxtzza

Figure 5.3: CID Format Figure
[124]


QmPo…GYvX

Protobuf UnixFS View on IPFS Gateway

CID QmPoMMxLZr2mrBdGo67bZj6B5Nfpay1qBB2eGWTBooGYvX
SIZE 2 MB
LINKS 9
DATA

```
▸Object {type: "file", data: null, blockSizes: Array[9]}
```

	PATH	CID
0	Links/0	QmPRLHi7kZiwfc686AosA4BVSAB9j5cdW8183e…
1	Links/1	QmePDwwzEetdNJBXuQkeVNdMNxKihg6oqYy4FT…
2	Links/2	QmcabnfZN7C2iKnmR1fcLtRTbCP9Rt8np2jrfa…
3	Links/3	QmTCDzMaou8JxTzHVh1bEKqe16XhyRL38z76zH…
4	Links/4	QmUZom8hDcMFoHwzS8wSVznU72JkojNWaXzKh1…
5	Links/5	QmQkfM7no5rHd96Z5roetxq3ufbux2x1UKWpJ5…
6	Links/6	QmUvUPBvaTXfhcjBxnG7B1ZXdYT4xTeBME7Qcr…
7	Links/7	QmUGKHmGYKrYCr8b39gTYxSBXBbwcsN7icqSdE…
8	Links/8	QmcRjoYzmfcKz1mTPjbg1RZgKyRNpGdP1ddBJ2…

CID INFO
QmPoMMxLZr2mrBdGo67bZj6B5Nfpay1qBB2eGWTBooG…
base58btc - cidv0 - dag-pb - sha2-256-256-15…
BASE - VERSION - CODEC - MULTIHASH

MULTIHASH

```
0x122015b2c1032654e4736b809e92da8cdbf1
      c492af0aab59f4a26b92c5c878a1f5f4
                            HASH DIGEST

0x12 = sha2-256
0x20 = 256 bits
```

Figure 5.4: CID Format Figure
[125]

* **Files:** Versioned file system hierarchy inspired by Git. The model consists of:
  1. block: a variable-size block of data.
  2. list: a collection of blocks or other lists.
  3. tree: a collection of blocks, lists, or other trees.
  4. commit a snapshot in the version history of a tree.
  5. **Naming:** IPFS uses the naming scheme from A self-certifying File System (SFS) [126] , a distributed file system that doesn't require special permissions for data exchange. It gives away to construct self-certified names, in a cryptographically assigned global namespace, that are mutable. Without it, all communication of new content must happen off-band, sending IPFS links.

5.3.3 How it Works?

It could be divided into two main steps :

1. **Joining the network as node:** To join the IPFS network, nodes generate identity to be identified by using a Public Key Infrastructure (PKI) in the following order :
   * A node generates a PKI key pair (public key and private key).
   * Hash the public key
   * The resulting hash is the NodeId
   * Upon first connecting, peers exchange public keys, and check: hashother.publi
     equal other.nodeId If not, the connection is terminated.
2. **Upload and fetch data:**
   (a) **Data upload** :
      * Data is chunked into smaller chunks, the IPFS default chunker is 256KB block size, then stored locally on the user's machine.
      * The network is notified and DHT is updated with the newly added data.

   (b) **Data download** :

- A node sends a fetch request with data CID.
- Routing protocol locates the addresses of nodes that they store the data segments
- Nodes download data from these nodes into their cache memory.

5.3.4 Availability

5.3.4.1 Garbage Collection

IPFS nodes store data in their case memory, but over time they reach full memory capacity and Nodes need to clear out previously cached resources to make room for new resources. Hence IPFS implements Garbage Collection (GC), the process within each IPFS node of clearing out cached files and blocks.

5.3.4.2 Pinning service

No guarantee in IPFS that the data will continue to be stored. If the user wants to ensure that content is retained over the long term. He should pin it to his local storage. ”Pinning” a CID tells an IPFS server that the data is important and mustn’t be thrown away. Hence, the garbage collector ignores it. This process typically called ”Pinning Service”.

5.3.5 Security

5.3.5.1 Network Attacks

[127] One of the most design issues in peer to peer network is its liability to some attacks that try to disrupt the service like Sybil and DDos attack, IPFS tries to mitigate this issue through the following

- Expanding on already implemented DHTs increases the IPFS security against Sybil and DDoS attacks.
- Using SFS to secure the data traffic, enable nodes to certify each other to avoid interacting with malicious nodes.

5.3.5.2 Data Integrity

IPFS by design provides end to end integrity .

5.3.5.3 Data Confedentiality

No built-in feature in IPFS for data confidentiality. Users can encrypt data before uploading it to IPFS. Some applications which are developed on top of IPFS provide end to end encryption as a feature.

5.3.5.4 Data leakage

In data upload, data is segmented and cryptographically hashed, no built-in function to prevent an attacker with the cryptographic hash of certain segments to obtain these certain segments and any other segments linked to them.

5.3.5.5 Pricing
Free regardless of the size of data.

5.3.5.6 Replication
When a user uploads data IPFS protocol partitions such data into segments stores them locally on the node's machine. Data will remain in the node-local machine until someone requests it. When data is requested, it's replicated in the local nodes that have requested that data.

5.3.5.7 Scalability
IPFS has Reached 100,000s of nodes in Apr 2020. In our IPFS practical implementation, our local node's peers ranged from 390-440 within the first 5 minutes as will be shown in the lab figures.

5.4 Filecoin

According to [128] , Filecoin is a decentralized storage network that turns cloud storage into an algorithmic market. The market runs on a blockchain with a native protocol token FIL (also called "Filecoin") on top of IPFS. It is developed by Protocol Labs, a U.S.-based company founded by Juan Benet. The FileCoin's ICO was launched in August 2017 with a fundraising goal of 40,000,000 USD, within 30 minutes, over 200 USD million was raised and In September the token sale ended achieving 257,000,000 USD (643 percent) of their fundraising goal. Filecoin is still in testnet, an alternative network to be used for testing, and will launch to miannet, the actual real network, by August 2020[]. Filecoin network employs Proof-of-Spacetime and Proof-of Replication [129] [130] to guarantee that miners have correctly stored the data they committed to store. Thus providing fixes to the inevitable drawbacks in IPFS such as instability and incentive mechanisms. Filecoin uses Expected Consensus (EC) [131] as a consensus algorithm in creating a new block. CE is a probabilistic Byzantine fault-tolerant consensus protocol, it operates by running a leader election every round. In each round, on expectation, one participant is eligible to submit a block.

Features that shared between filecoin and IPFS :

∗ Using IPLD for blockchain data structures.
∗ Using libp2p for peer communication and connection establishment
∗ CIDs in both Filecoin and IPFS share a hashing specification
∗ Using graphsync [133] for transferring data between nodes, to enable direct transfers between IPFS and Filecoin nodes in the future.

5.4.1 Core Components

Filecoin protocol consists of four novel components:

1. **Decentralized Storage Network:** an abstraction for a network of independent storage providers to offer storage and retrieval services. in

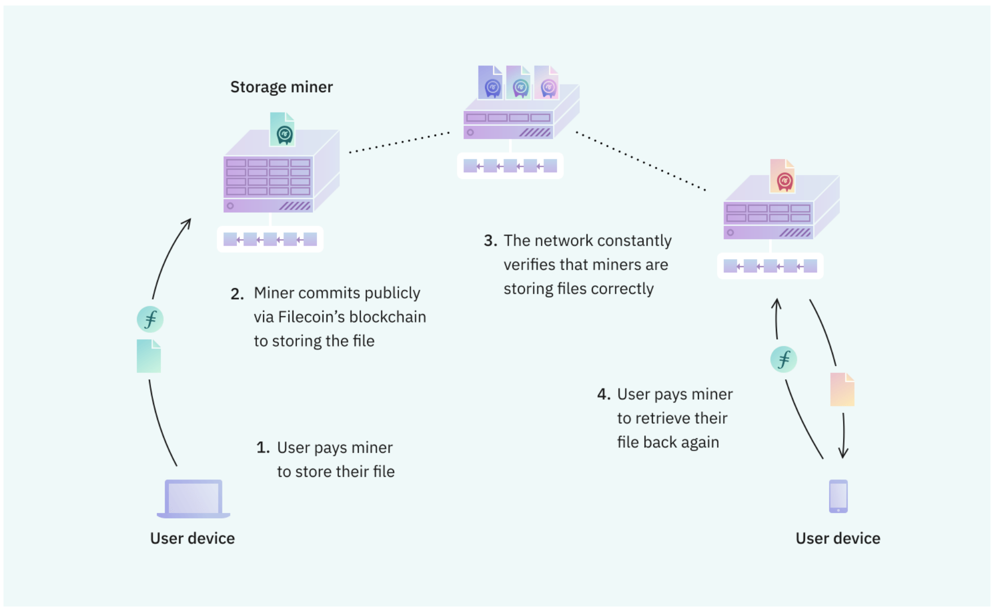
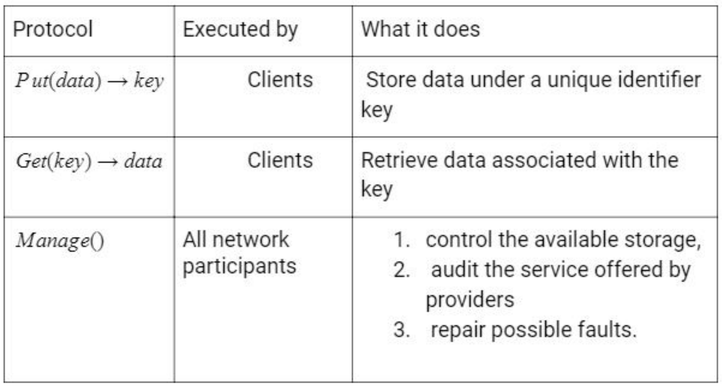

Storage miner
2. Miner commits publicly via Filecoin's blockchain to storing the file
3. The network constantly verifies that miners are storing files correctly
4. User pays miner to retrieve their file back again
1. User pays miner to store their file
User device
User device

Figure 5.5: Filecoin life cycle
[132]

which tuple of protocols (Put, Get, Manage) run by both Miners and clients.



Protocol	Executed by	What it does
$Put(data) \to key$	Clients	Store data under a unique identifier key
$Get(key) \to data$	Clients	Retrieve data associated with the key
$Manage()$	All network participants	1. control the available storage, 2. audit the service offered by providers 3. repair possible faults.

Figure 5.6: Put and Get functions

2. **Novel Proofs-of-Storage:** two novels Proofs-of-Storage: Proof-ofReplication: to prove that data has been replicated to its own uniquely dedicated physical storage. Proof-of-Spacetime : to prove that storage providers have stored some data throughout a specified amount of time.
3. **Verifiable Markets:** Verifiable markets for storage requests and retrieval requests to ensure that payments are performed when a service has been provided correctly.
4. **Useful Proof-of-Work:** Instead of spending wasteful computation to mine blocks, Miners only need to store data in the network. The voting power is not by the computational power he spent to solve the puzzle, instead, it is proportional to their storage currently in use in relation to the rest of the network.

5.4.2 Participants

Participants in filecoin network are classified into three types:

1. Clients who are willing to pay for storing or retrieving data.
2. Storage miners who get tokens for offering storage space. To become Storage Miners, users must pledge their storage by depositing collateral proportional to it via Manage.PledgeSector. This process guarantees the miners' reliability as the network penalizes the Storage Miners by taking part of their collateral if they fail to prove the data availability and integrity. Storage Miners are eligible to be elected to mine new blocks and receive the mining reward for creating a block and transaction gas for the transactions included in the block.
3. Retrieval Miners who get tokens for retrieving data from storage miners to clients. They don't have to store pieces through time nor generate proofs of storage like Storage Miners.

5.4.3 How it Works?

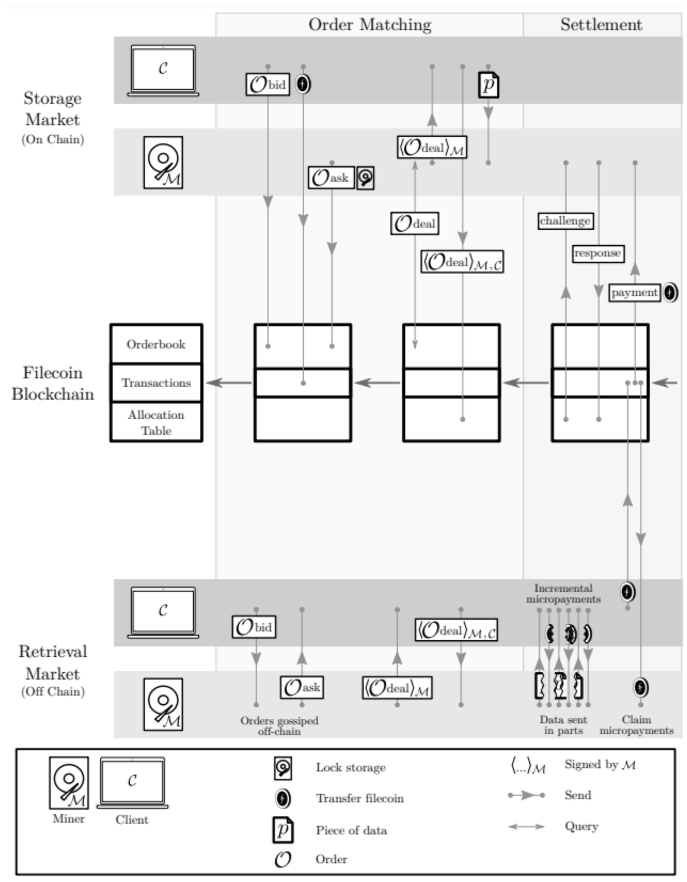


Figure 5.7: Filecoin stack
[134]

5.4.3.1 Storing data

At a high level; a client can choose his preferred trade-off between cost, redundancy, and speed by selecting the miner whose storage offer is best suited for his needs and submits a bid order, statement of intent to request or offer a service, to the Storage Market order book, a list of orders, via Put.addOrder protocol. Storage Miners respond to Put.addOrder requests and commit to store the client's data for a specified time with ask order. Storage negotiation with any miner is done by applications that implement filecoin protocol. When their orders match, they jointly create a deal order that commits both them (client and storage minner) to the exchange. The deal is propagated to the network and added to the orderbook. At this point, the client starts sending data segments to the Storage Miner, the Network assigns (AssignOrders) the data to that miner, and makes a note of it in the allocation table, a data structure that keeps track of pieces and their assigned sectors. When a Storage Miner sector, some disk space that a Storage Miner provides to the network, is filled with client's data pieces, a piece is some part of data that a client is storing in the DSN, the sector is sealed. Sealing, a slow sequential operation, is for transforming the data in a sector into a replica, a unique physical copy of the data associated with Storage Miner's public key. During the Proof-of-Replication Sealing, the Sealing operation is necessary. Worth mentioning that all transactions are done in public on-chain and the network verifies them. At any time, the client can verify that his files are being stored correctly by looking at proofs on Filecoin's blockchain. In each newly added block, the Network checks if the required proofs for each assignment are present, valid, and act accordingly. Storage Miners respond by submitting Proofs-of-Spacetime to prove that they are storing the data through time. Filecoin has three scenarios :

1. In case of invalid or missing proofs, the Filecoin protocol penalizes Storage Miners by losing part of their collateral.
2. If large parts of proofs are missing or invalid, the Storage Miner is considered faulty and the network settles the order as failed and reintroduces a new order for the same piece into the market.
3. If every Storage Miner storing this piece is faulty, the piece is lost and the client gets refunded.

When order's time is expired or funds run out, the Network processes the payments to the storage miners and removes the orders as long as the service was correctly provided.

5.4.3.2 Retrieving data

Retrieving data is done through the retrieval market where the client finds the Retrieval Miner who is willing to serve the required pieces and directly exchange the pieces after they both settle on the price. Retrieval market is operating off-chain, outside the filecoin blockchain network to mitigate network bottleneck for fast retrieval requests. Hence, they both have to announce their orders via gossiping, p2p communication protocol. Thus, to secure the exchange without relying on any trusted parties since the network, trusted party, is not witnessing the exchange, the Retrieval Miners split their data into multiple parts and clients pay for each part they receive via payment channels. The cycle starts with Retrieval Miners gossip their newly created ask orders and clients gossip their created bid orders. Both are submitting Get.addOrders. No commonly shared orderbook, so users add orders to their own orderbook's view whenever they receive orders with no committed resource. When their orders match, they jointly create a deal order and add it to the Get.OrderBook. Miner starts sending each piece via Get.SendPiece and the client responds and acknowledges delivery by sending a signed receipt via Get.ReceivePiece which the miner presents to the blockchain to get his rewards. Once the retrieval operation is done successfully, pieces are delivered and payment is sent, both (client and miner) remove their orders from the order books.

5.4.4 Availability

Filecoin guarantees data availability in a self-healing mechanism. The Network checks, in every newly added block, whether the required proofs for each assignment are present, valid, and act accordingly. Filecoin has three scenarios :

1. In case of invalid or missing proofs, the Filecoin protocol penalizes Storage Miners by losing part of their collateral.
2. If large parts of proofs are missing or invalid, the Storage Miner is considered faulty and the network settles the order as failed and reintroduces a new order for the same piece into the market.
3. If every Storage Miner storing this piece is faulty, the piece is lost and the client gets refunded.

5.4.5 Security

FileCoin inherits IPFS security mechanism as well as employs Proofs-of-Storage to mitigate three types of attacks from malicious miners :

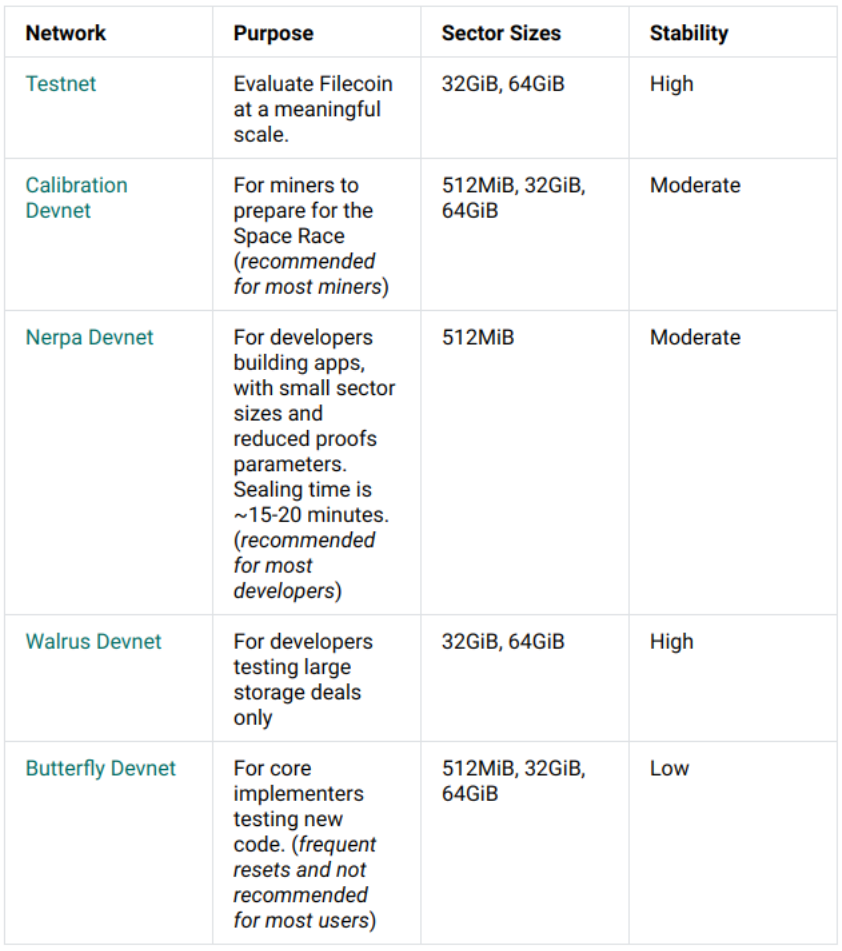
1. **Sybil Attacks:** creating multiple Sybil identities to store multiple copies of data, but storing the data only once.
2. **Outsourcing Attacks:** committing to store more data than the amount they can physically store.
3. **Generation Attacks:**Claim they are storing a large amount of data while they are not.

5.4.6 Pricing

Determined by supply and demand and negotiable between miners and users in an open market.

5.4.7 Scalability

Filecoin inherits the scalability features from IPFS. Several Filecoin networks for various testing, benchmarking, and optimization needs are built. The following table illustrates each as they mentioned in their documentation [135].



Network	**Purpose**	**Sector Sizes**	**Stability**
Testnet	Evaluate Filecoin at a meaningful scale.	32GiB, 64GiB	High
Calibration Devnet	For miners to prepare for the Space Race (*recommended for most miners*)	512MiB, 32GiB, 64GiB	Moderate
Nerpa Devnet	For developers building apps, with small sector sizes and reduced proofs parameters. Sealing time is ~15-20 minutes. (*recommended for most developers*)	512MiB	Moderate
Walrus Devnet	For developers testing large storage deals only	32GiB, 64GiB	High
Butterfly Devnet	For core implementers testing new code. (*frequent resets and not recommended for most users*)	512MiB, 32GiB, 64GiB	Low

By exploring File Scan Website in Calibration network, we have noticed that miners distribution is across the world but there is mainstream adoption in China figure file coin peer and the efficiency in the latest 24 hour of Total Network is 13.46TB/H.

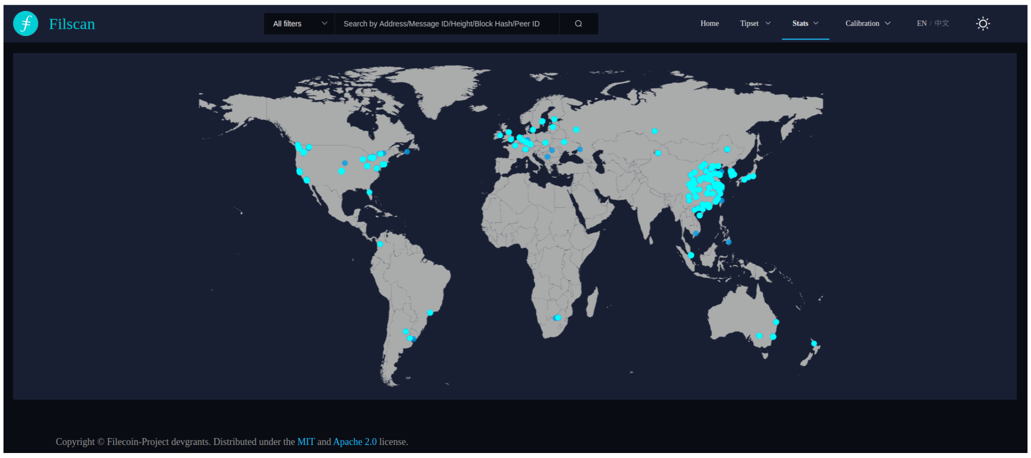

Filscan
All filters
Search by Address/Message ID/Height/Block Hash/Peer ID
Home
Tipset
Stats
Calibration
EN / 中文
Copyright © Filecoin-Project devgrants. Distributed under the MIT and Apache 2.0 license.

Figure 5.8: Filecoin peers distribution
[59]

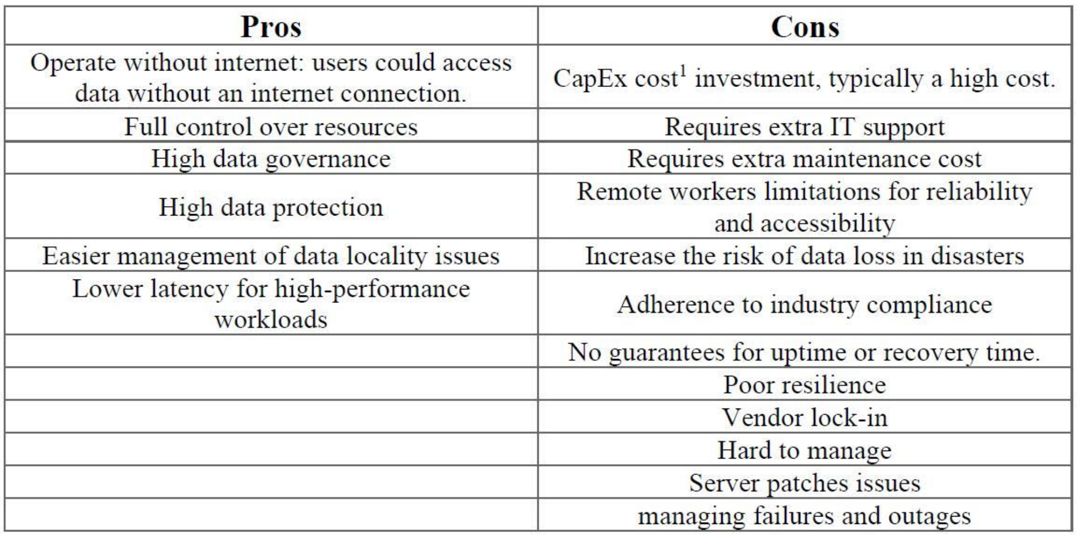
6 Conclusion

6.1 Analysis

Through the past chapters, we have explored various storage models with different topologies and capabilities, in this chapter we are articulating the research outputs. This chapter is divided into four parts. In the first part, we are concluding the main pros and cons of each. In the second part, we are giving a summarized technical comparison between each technology. The third part contains the survey result and in the last part, we are giving our recommendations for businesses based on some proposed use cases.

6.2 Pros and Cons of each storage model

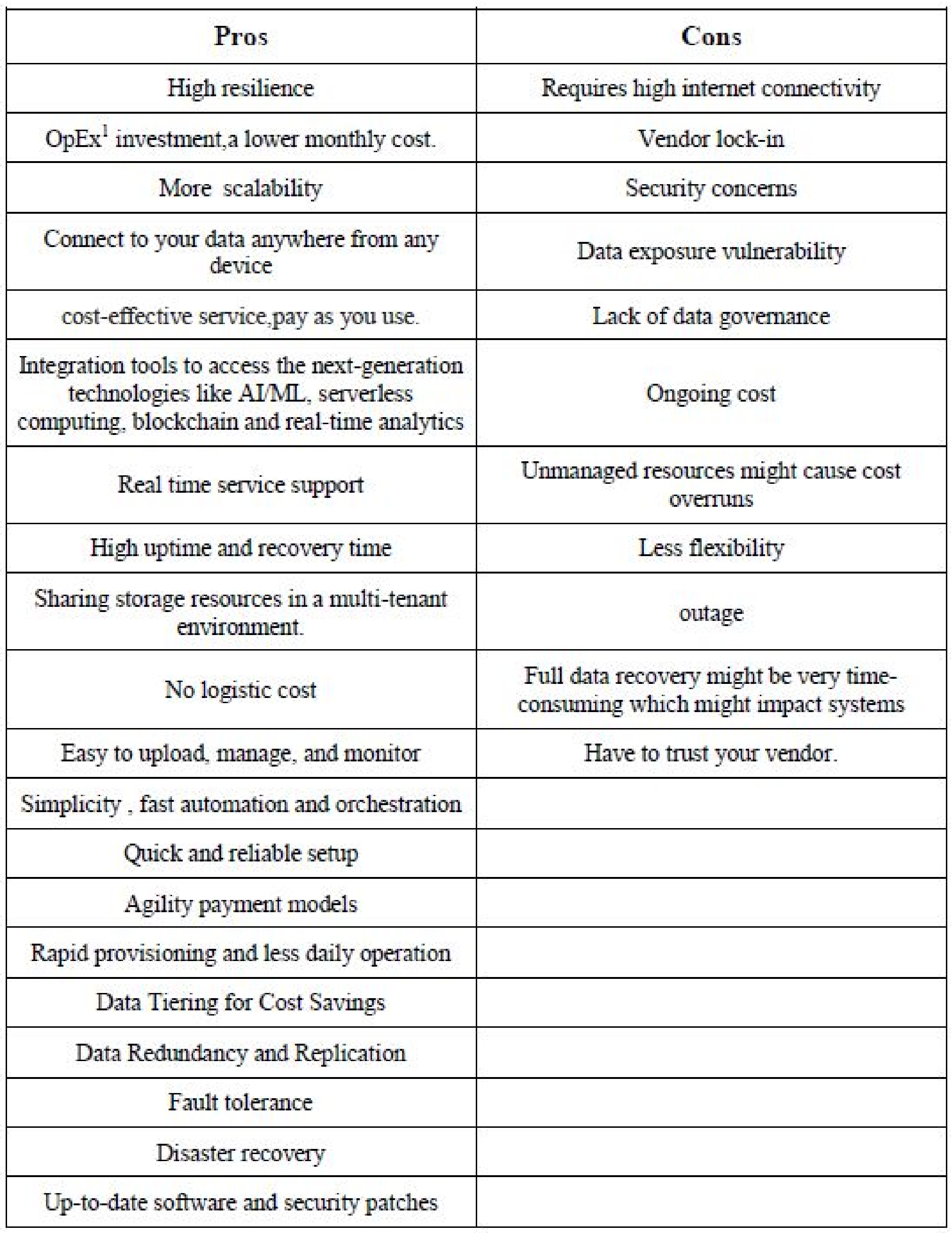
6.2.1 On-premises storage



Pros	Cons
Operate without internet: users could access data without an internet connection.	CapEx cost[1] investment, typically a high cost.
Full control over resources	Requires extra IT support
High data governance	Requires extra maintenance cost
High data protection	Remote workers limitations for reliability and accessibility
Easier management of data locality issues	Increase the risk of data loss in disasters
Lower latency for high-performance workloads	Adherence to industry compliance
	No guarantees for uptime or recovery time.
	Poor resilience
	Vendor lock-in
	Hard to manage
	Server patches issues
	managing failures and outages

Figure 6.1: On-premises storage pros and cons

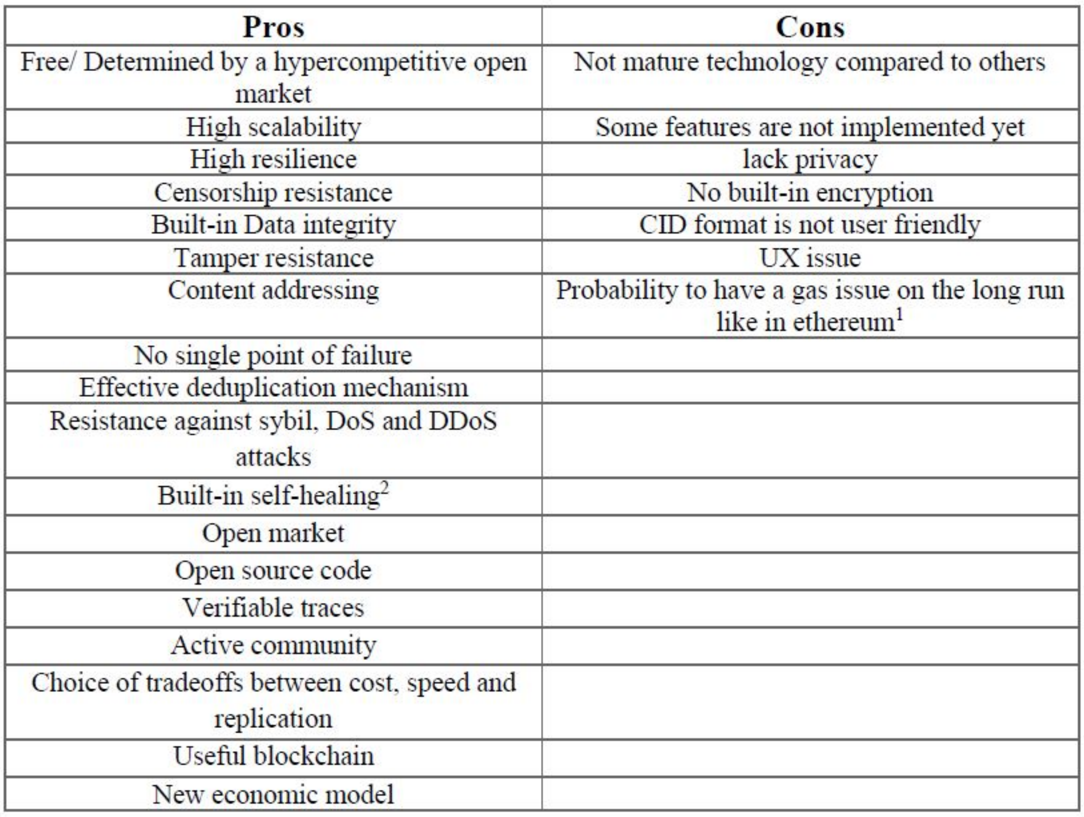
6.2.2 Cloud based storage



Pros	Cons
High resilience	Requires high internet connectivity
OpEx[1] investment,a lower monthly cost.	Vendor lock-in
More scalability	Security concerns
Connect to your data anywhere from any device	Data exposure vulnerability
cost-effective service,pay as you use.	Lack of data governance
Integration tools to access the next-generation technologies like AI/ML, serverless computing, blockchain and real-time analytics	Ongoing cost
Real time service support	Unmanaged resources might cause cost overruns
High uptime and recovery time	Less flexibility
Sharing storage resources in a multi-tenant environment.	outage
No logistic cost	Full data recovery might be very time-consuming which might impact systems
Easy to upload, manage, and monitor	Have to trust your vendor.
Simplicity , fast automation and orchestration	
Quick and reliable setup	
Agility payment models	
Rapid provisioning and less daily operation	
Data Tiering for Cost Savings	
Data Redundancy and Replication	
Fault tolerance	
Disaster recovery	
Up-to-date software and security patches	

Figure 6.2: Cloud based storage pros and cons

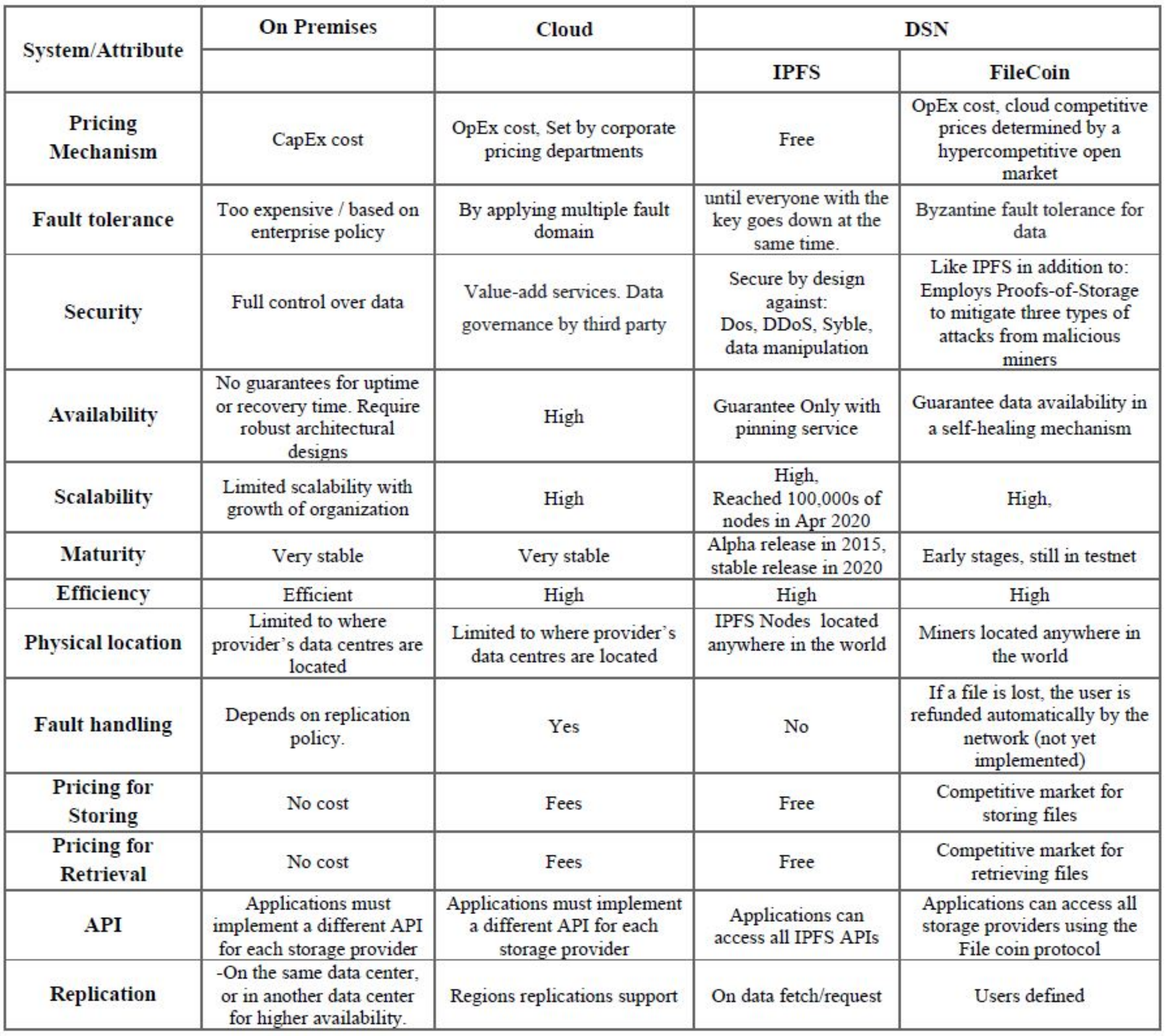
6.2.3 DSN



Pros	Cons
Free/ Determined by a hypercompetitive open market	Not mature technology compared to others
High scalability	Some features are not implemented yet
High resilience	lack privacy
Censorship resistance	No built-in encryption
Built-in Data integrity	CID format is not user friendly
Tamper resistance	UX issue
Content addressing	Probability to have a gas issue on the long run like in ethereum[1]
No single point of failure	
Effective deduplication mechanism	
Resistance against sybil, DoS and DDoS attacks	
Built-in self-healing[2]	
Open market	
Open source code	
Verifiable traces	
Active community	
Choice of tradeoffs between cost, speed and replication	
Useful blockchain	
New economic model	

Figure 6.3: DSN pros and cons

6.2.4 Summarized technical comparison



System/Attribute	On Premises	Cloud	DSN	
			IPFS	FileCoin
Pricing Mechanism	CapEx cost	OpEx cost, Set by corporate pricing departments	Free	OpEx cost, cloud competitive prices determined by a hypercompetitive open market
Fault tolerance	Too expensive / based on enterprise policy	By applying multiple fault domain	until everyone with the key goes down at the same time.	Byzantine fault tolerance for data
Security	Full control over data	Value-add services. Data governance by third party	Secure by design against: Dos, DDoS, Syble, data manipulation	Like IPFS in addition to: Employs Proofs-of-Storage to mitigate three types of attacks from malicious miners
Availability	No guarantees for uptime or recovery time. Require robust architectural designs	High	Guarantee Only with pinning service	Guarantee data availability in a self-healing mechanism
Scalability	Limited scalability with growth of organization	High	High, Reached 100,000s of nodes in Apr 2020	High,
Maturity	Very stable	Very stable	Alpha release in 2015, stable release in 2020	Early stages, still in testnet
Efficiency	Efficient	High	High	High
Physical location	Limited to where provider's data centres are located	Limited to where provider's data centres are located	IPFS Nodes located anywhere in the world	Miners located anywhere in the world
Fault handling	Depends on replication policy.	Yes	No	If a file is lost, the user is refunded automatically by the network (not yet implemented)
Pricing for Storing	No cost	Fees	Free	Competitive market for storing files
Pricing for Retrieval	No cost	Fees	Free	Competitive market for retrieving files
API	Applications must implement a different API for each storage provider	Applications must implement a different API for each storage provider	Applications can access all IPFS APIs	Applications can access all storage providers using the File coin protocol
Replication	-On the same data center, or in another data center for higher availability.	Regions replications support	On data fetch/request	Users defined

Figure 6.4: Summarized Technical Comparison

6.3 Our Recommendation

So at the end, What is better for your company; on-premises or cloud or using DSN? Many established companies wonder if it is worth it to transition out of their on-premises technological infrastructure and move on to the cloud or to implement it on DSN. In contrast, several newer companies wonder if they should invest their early capital in on-premises systems. Each technology might have its own enthusiasts who believe in its capabilities and potential impact, have good marketing strategies to make higher business profits or have a long success history for decades with giant companies behind with well-established networks and businesses. Each party might say that his solution fits your needs. Among the endless claims from each party, lots of misleading decisions were made, lots of money were lost and at the end, the business might fail and we might lose an innovative idea that might make our life much better.

Generally, to choose which option is right for your company, it is important to think about the needs of your business. There are trade-offs to whatever option you choose, so you should be fully informed before you decide how many on-premises or cloud services or even block chain technology you include at your company.

In this part, we will try to articulate the potential business use cases and evaluate how much each model might fit. The order of model suggestion is gradually from best to least suit. So, According to your business needs :

∗ **less capital investment:**Some DSNs give you free storage service others give you cloud competitive prices determined by a hypercompetitive open market, thus DSN is recommended to be your first choice. The second choice would be definitely using Cloud as opEx investment model, with pay as you use/go feature, you only pay for the resources that you are using. While in your first set up on-premises storage, which is capEx investment, you'll have to invest a significant amount of capital to purchase the servers and other pieces of hardware to get started. For companies just trying to get off the ground, this level of capital investment can be a huge disadvantage. Along with purchasing the equipment, you'll also need to devote time and money to make sure it's properly installed.

∗ **Provide greater security:**Higher confidentiality: on-premises would be your first and best option. Unlike cloud-storage, which is more vulnerable to third parties and prying eyes, on-premises storage is completely restricted from anyone other than authorized personnel. On-premises servers are not accessible to those outside the network, as they are not storing the data online. For companies who handle sensitive data, like those in the financial industry, on-premises storage may be a preferred option. Cloud storage with encryption at transit and at rest to protect your data from data breach vulnerability which happened from time to time .DNS could be your third option by considering using platforms that have a built-in encryption mechanism or encrypt your data before uploading it. Choosing the strongest encryption algorithm would protect your data from some attacks to compromise it.

∗ **Higher Integrity:** DSN is tamper-resistance by design and It could give you a higher level of integrity. Thus DSN is your first option. Cloud implements different data integrity checking algorithms; which eliminates the third party auditing. Also on premises, it is more secure and more manageable so there is better chance of data integrity.

∗ **Higher scalability with less efforts:** If your company scales up and needs more storage space or other capabilities, DSN is scalable by design and would give you the highest scalability compared to the other alternatives with no/less cost. Cloud could give you more scalability over on-premises. it's more difficult to scale your on-premises servers quickly. Unlike cloud-storage, where companies can simply pick a more expansive plan with a click, on-premises storage requires you to install new hardware and devote manpower to building the new systems.

∗ **High-Frequency Data**: If your data might be requested frequently, you could get the benefits of DSN robust design and with no cost. DSN will scale as requests grow; the more frequently your data is being requested, the more available it will be. Cloud could be your second choice as it could scale fast as needed but with extra cost and less efficiency compared to DSN. On-premises would be your worst choice in terms of cost , time, management and efficiency.

∗ **Secure Disaster Recovery:** If your business needs disaster recovery service, your first choice would be Cloud. Cloud provides Disaster Recovery as a Service (DRaaS). In terms of cost, time , replication and maturity, Cloud is better than the other alternatives. DSN comes as a second choice as it lacks maturity. On-premises is your last choice because of time and cost to build the infrastructure.

∗ **Offline operation:** One of the major upsides to on-premises storage is that it does not require users to have an internet connection to access data. Though most businesses rely on the internet to conduct business, there's always a fear that the loss of a connection could harm productivity and make it impossible to access crucial data. On-premises servers will provide you with an internal network that is accessible anytime, no matter your internet connection. Neither cloud nor DSN will fit your business in this case.

∗ **Lower monthly internet costs:** If your business does not rely on any internet services, you may not need to pay for such a high-speed connection. For those with on-premises storage, the need for a strong connection with fast download speeds is reduced even further. DNS requires lower connection speed compared to cloud especially with dealing with large files. Based on your needs, you may not have to pay for a more expensive internet plan if you don't have to access the cloud to view files.

∗ **Control over server hardware:** Some companies prefer having dedicated servers within their building to handle all their needs. In this case, on-premises is your first option. The second choice would be using the cloud but a concern might arise; if you ask a cloud storage company to upgrade their storage plan or add new features, the company can simply do the upgrades themselves. Potentially, being able to modify the server's hardware can give savvy companies greater flexibility and customization for their storage needs. DSN is not applicable here.

* **less IT support:** Both DSN and Cloud would be your best suit in this case. Given that DSN in terms of price is less than cloud, we think DSN could be your first choice then Cloud as the second choice. If you decide you want to use on-premises storage, you'll also need to have IT staff to maintain and manage your servers. Hence you have to hire new staff members or devote more of your current staff's time to maintaining the servers. This extra support can add to your costs and reduce the efficiency of your IT department as they will have increased responsibilities associated with the on-premises servers.
* **Stable technology:** If you are searching for technical stability to avoid unexpected technical issues, Both On-premises and Cloud are very stable. We recommend Cloud as the first choice as it opEx investments and provides lots of on-demand robust services and tools. The second choice is definitely on-premises with high stability and long history. DSN is a new technology and not mature enough. Picking DSN as an option comes with the tradeoff of immaturity issues.

6.4 Survey output

In August 2020, We conducted a survey with a purposive sampling, focusing on senior to managerial role-based with hands-on experience in on-premises and cloud check Variety of experience level Figure below . With average years of experience of 8.1, minimum 3 years and 18 years Maximum. Our ultimate goals are to validate our research hypothesis and explore the local market through in person distributed surveys. We used both open and close question types based on the kind of information we needed to know.

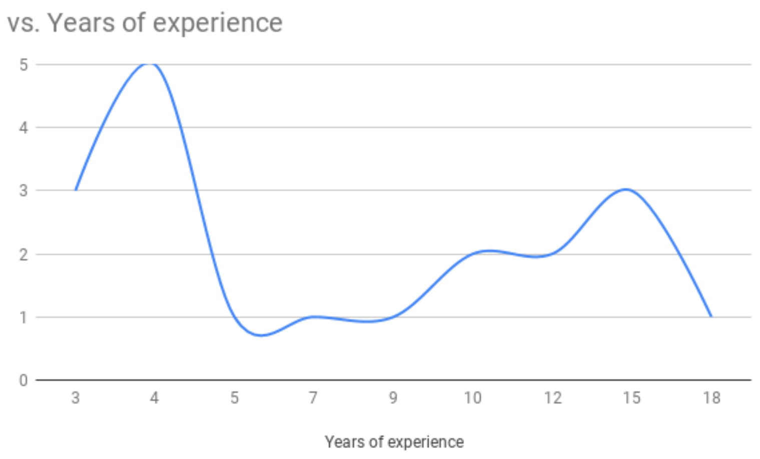


Figure 6.5: Variety of experience level Figure

The survey identified several key findings as follows:

1. 68.4 percent has experience working on storage on-premises .

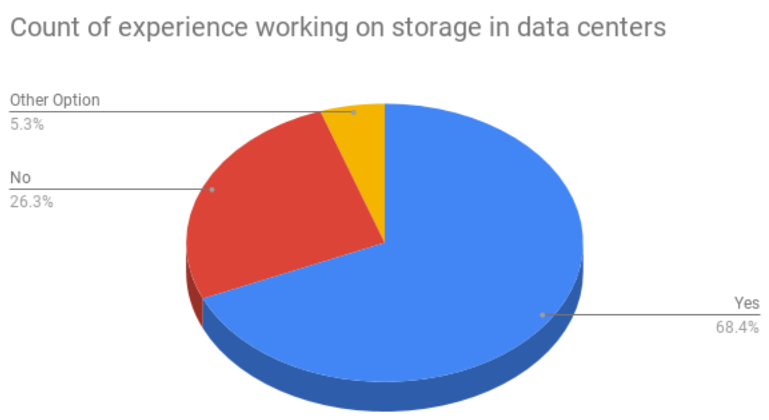


Figure 6.6: Count of people working on storage in data center.

2. 47.1 percent has encountered vendor lock-in in upgrading storage on-premises ,while 21.1 percent has encountered vendor lock-in in migration on Cloud.

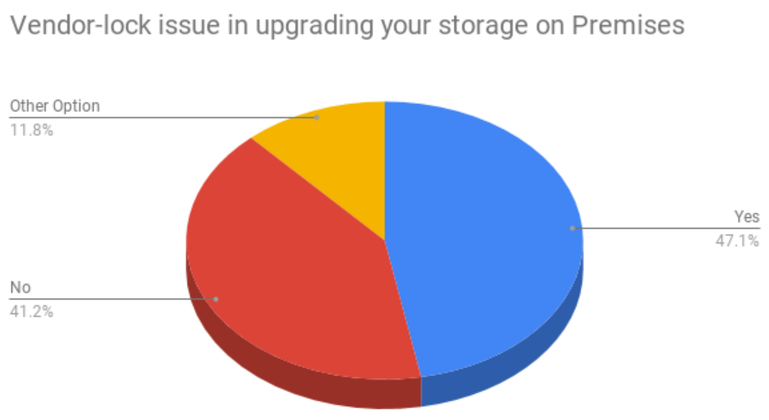


Figure 6.7: Count of people who faced vendor lock-in issue in upgrading storage on-premises.

3. The overall satisfaction rate using storage on-premises was 4 , while the overall satisfaction rate using storage on Cloud was 4.1.
4. 46.6 percent was using Azure , 46.6 percent was using AWS , 33.3 percent was using google cloud and 40 percent was using other cloud providers .
5. 47.4 percent was already aware of the DSN. Worth mentioning, only 0.5 percent of them could mention a correct DSN network which means that this result is misleading.
6. 42.1 percent was interested in mining with their legacy storage to get revenue.

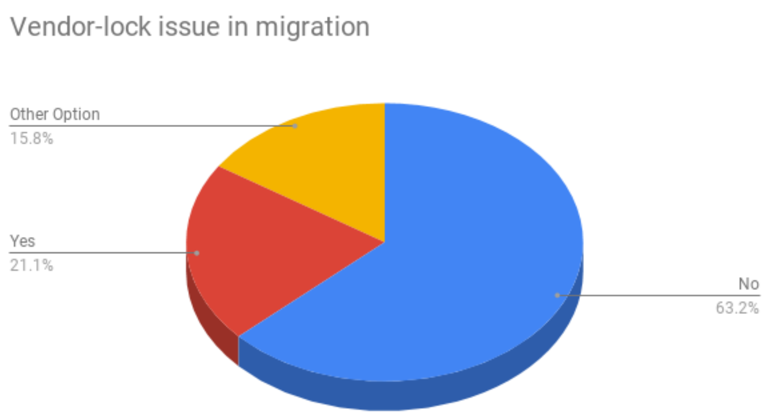


Figure 6.8: Count of people who faced vendor lock-in issue in migration.

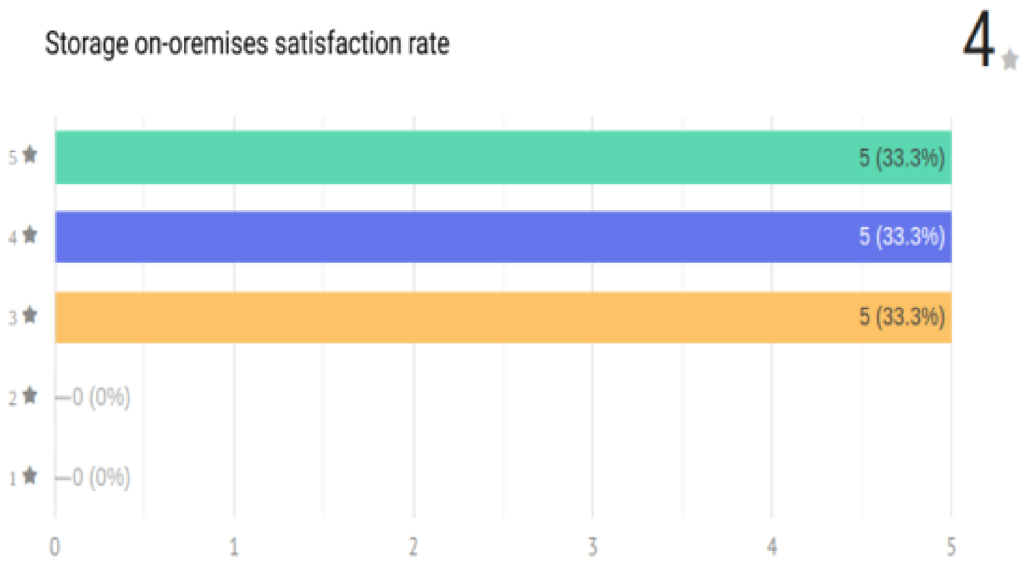


Figure 6.9: satisfaction rate using storage on-premises

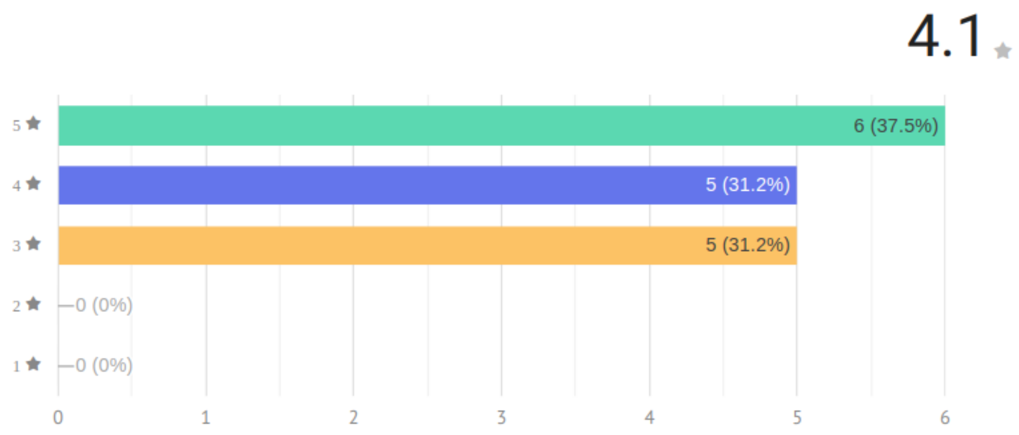


Figure 6.10: satisfaction rate using storage Cloud

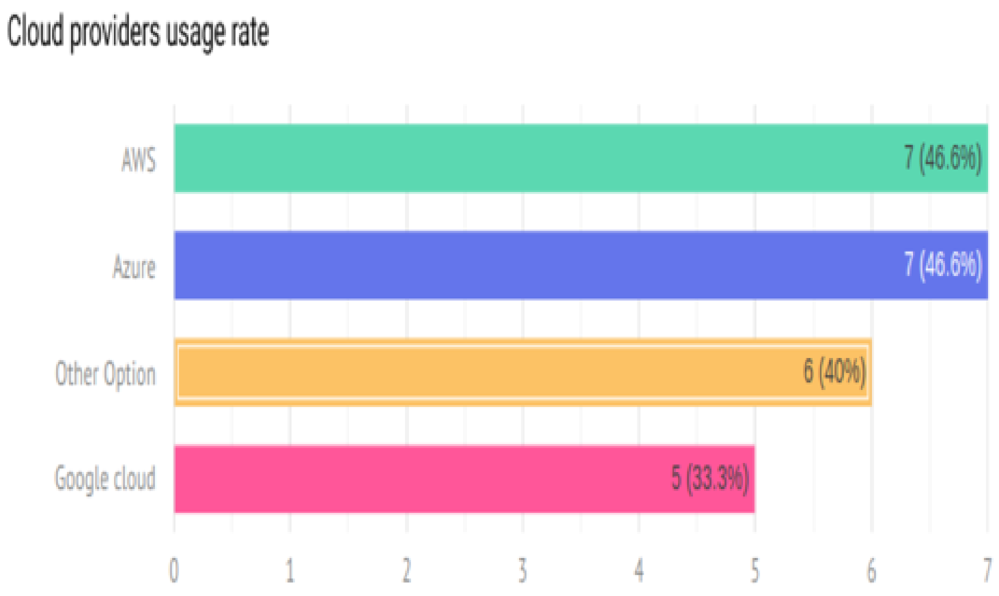


Figure 6.11: Cloud providers usage rate

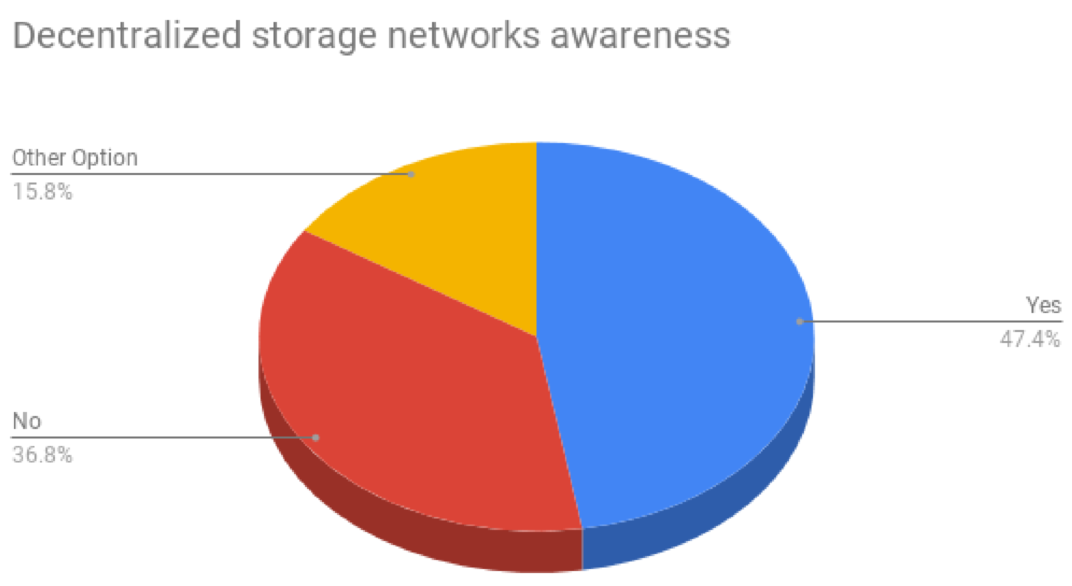


Figure 6.12: Decentralized Storage networks

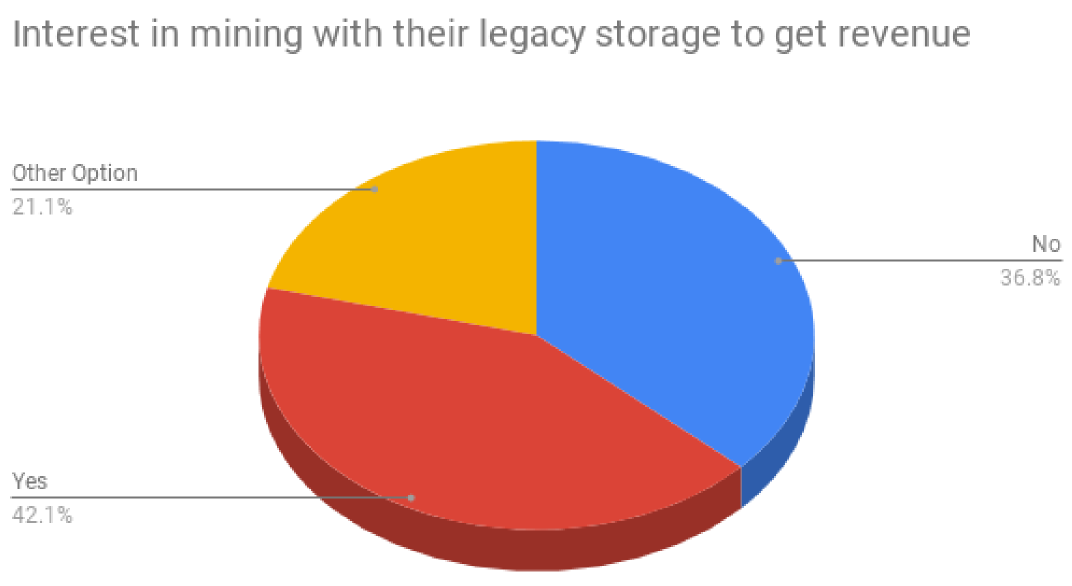


Figure 6.13: Number of people interested in mining with their legacy storage to get revenue

6.5 Conclusion

Through this paper, we have explored storage in different models; defining the state of the art of each technology and articulating opportunities, and challenges; technical and from business-wise. Our goal is to inform the reader of the latest updates, similarities, and differences to choose his trade-offs based on his needs and a good knowledge base. We believe that each model will remain, they are not competitive, regardless of the endless race, each has its unique value proposition that fits a real use case.

A Appendix A:ECS Object Storage

This appendix will show the ECS Connection Manager”Kemp” interface, and the support for different ECS deployment use cases including TLS Offloading to optimize performance and purpose built Templates to ensure desired configurations based on best practices.

A.1 ECS Connection Manager

A.1.1 Create Virtual Service

1. On the desktop double click Google Chrome to open the browser.

Figure A.1: Open Google Chrome

2. Click ECS Conn Mgr in the Bookmarks Bar.

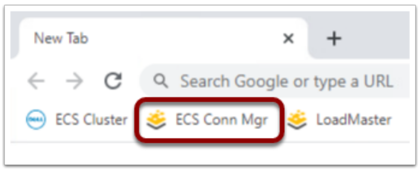


Figure A.2: ECS Conn Mgr

3. Enter Credentials,Navigate to SSL certificates.”A Certificate must be added on ECS Connection Manager to support the secure connections from the application accessing the storage meanwhile optimizing performance by enabling SSL/ TLS Offloading”.

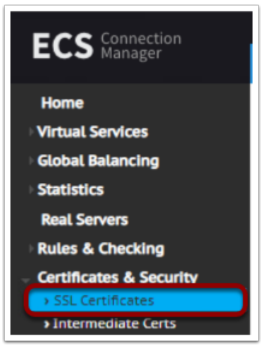


Figure A.3: SSL Certificates

4. Click Import Certificate.

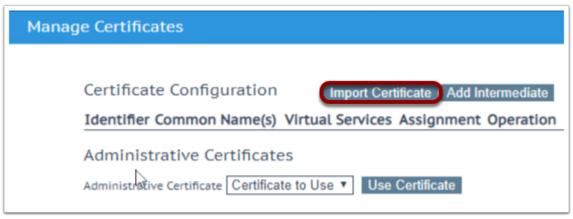

Manage Certificates
Certificate Configuration
Import Certificate
Add Intermediate
Identifier Common Name(s) Virtual Services Assignment Operation
Administrative Certificates
Administrative Certificate
Certificate to Use
Use Certificate

Figure A.4: Import Certificate.

5. Choose File,Navigate to Desktop and open the folder named Certificate,Select kemp.pfx and click Open.

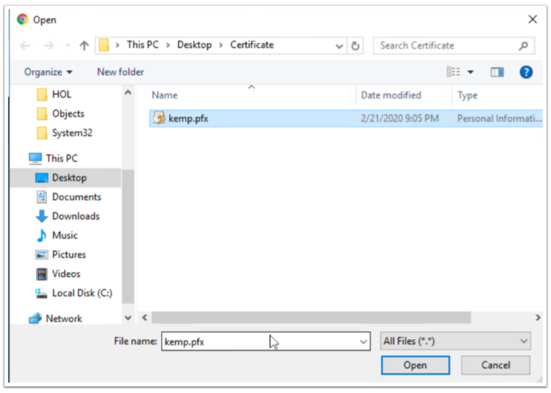
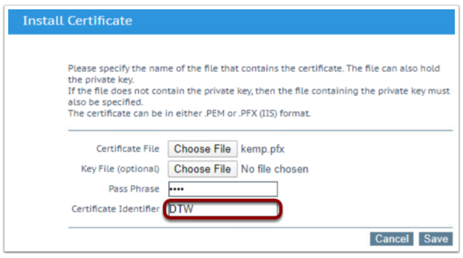


Figure A.5: Choose Kemp Certificate.

6. Enter password and certificate identifier.



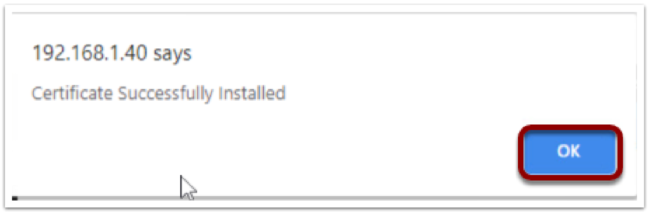
Install Certificate

Please specify the name of the file that contains the certificate. The file can also hold the private key.
If the file does not contain the private key, then the file containing the private key must also be specified.
The certificate can be in either .PEM or .PFX (IIS) format.

Certificate File: kemp.pfx
Key File (optional): No file chosen
Pass Phrase: ••••
Certificate Identifier: DTW

Figure A.6: password and certificate identifier.

7. Certificate installed successfully.

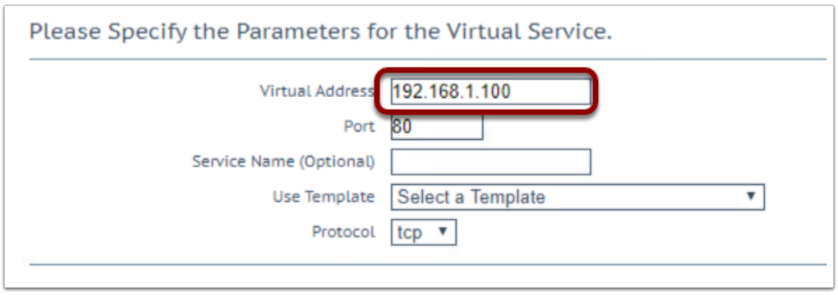


192.168.1.40 says

Certificate Successfully Installed

Figure A.7: Certificate installed successfully

8. Navigate to Virtual Services and Add New , and enter the following Virtual Service IP Address 192.168.1.100

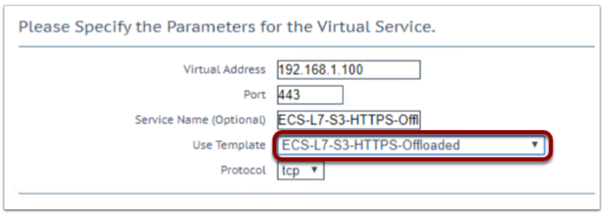


Please Specify the Parameters for the Virtual Service.

Virtual Address	192.168.1.100
Port	80
Service Name (Optional)	
Use Template	Select a Template
Protocol	tcp

Figure A.8: Enter Virtual Service IP Address

9. Select Template.

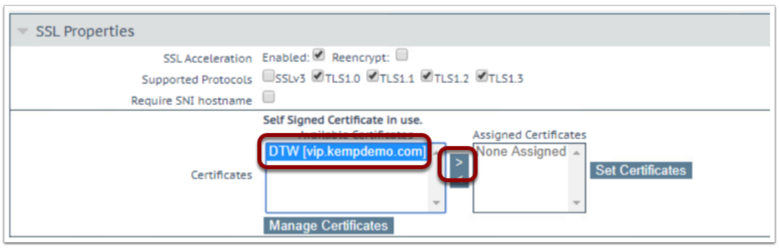


Please Specify the Parameters for the Virtual Service.

Virtual Address	192.168.1.100
Port	443
Service Name (Optional)	ECS-L7-S3-HTTPS-Offl
Use Template	ECS-L7-S3-HTTPS-Offloaded
Protocol	tcp

Figure A.9: Select Template

10. Select Certificate.

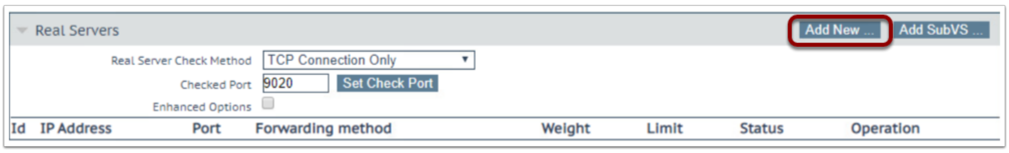


SSL Properties

SSL Acceleration	Enabled: ✓ Reencrypt: ☐
Supported Protocols	☐ SSLv3 ✓ TLS1.0 ✓ TLS1.1 ✓ TLS1.2 ✓ TLS1.3
Require SNI hostname	☐
Certificates	Self Signed Certificate in use. Available Certificates: DTW [vip.kempdemo.com] Assigned Certificates: None Assigned

Figure A.10: Select Certificate

11. Open real servers section, and add new
12. Add ECS NODE 1, and repeat this for Node2,3 and 4.

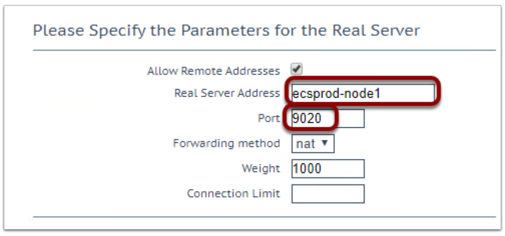


Real Servers

Real Server Check Method	TCP Connection Only
Checked Port	9020
Enhanced Options	☐

Id	IP Address	Port	Forwarding method	Weight	Limit	Status	Operation

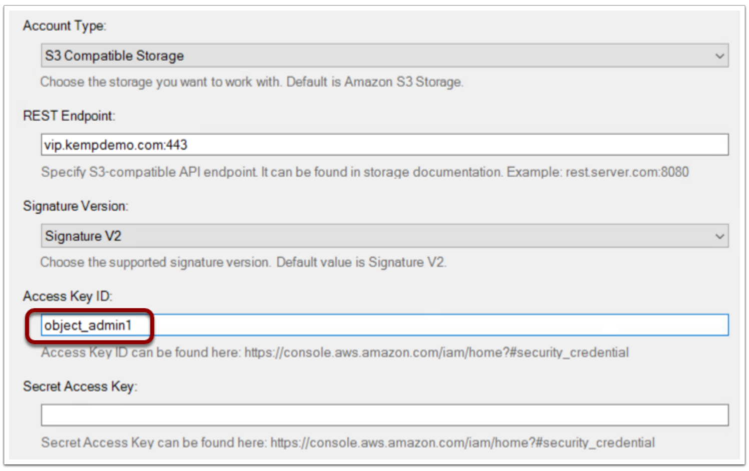
Figure A.11: Add real server


Please Specify the Parameters for the Real Server

Allow Remote Addresses ✓
Real Server Address ecsprod-node1
Port 9020
Forwarding method nat
Weight 1000
Connection Limit

Figure A.12: Add ECS NODE 1

Figure A.13: Open S3, add new account ECS connect


Account Type:
S3 Compatible Storage
Choose the storage you want to work with. Default is Amazon S3 Storage.

REST Endpoint:
vip.kempdemo.com:443
Specify S3-compatible API endpoint. It can be found in storage documentation. Example: rest.server.com:8080

Signature Version:
Signature V2
Choose the supported signature version. Default value is Signature V2.

Access Key ID:
object_admin1
Access Key ID can be found here: https://console.aws.amazon.com/iam/home?#security_credential

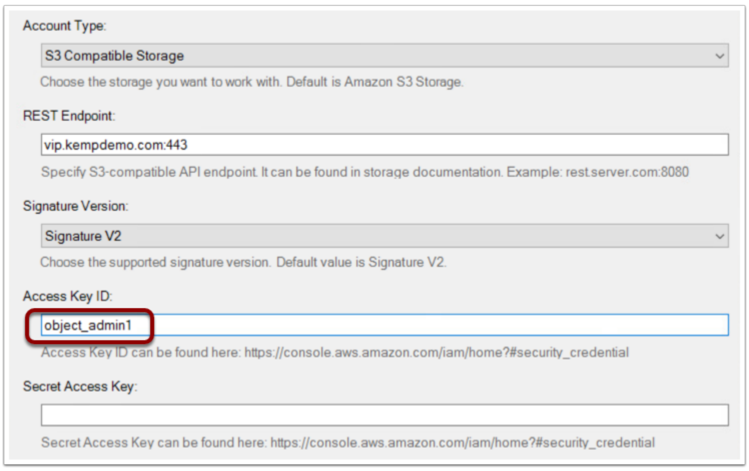
Secret Access Key:

Secret Access Key can be found here: https://console.aws.amazon.com/iam/home?#security_credential

Figure A.14: Add object id

A.2 Connect to ECS through ECS connection manager

1. Double click on S3.
2. Proceed to ECS web interface. An Object ID and Secret Access Key are provided by ECS and are required for authentication to the storage .
3. Save configuration.
4. After clicking Add new account in the previous step the S3 Browser will have a successful connection to the ECS Storage. The ability to view the available bucket(s) verifies an established connection.



Account Type:

S3 Compatible Storage

Choose the storage you want to work with. Default is Amazon S3 Storage.

REST Endpoint:

vip.kempdemo.com:443

Specify S3-compatible API endpoint. It can be found in storage documentation. Example: rest.server.com:8080

Signature Version:

Signature V2

Choose the supported signature version. Default value is Signature V2.

Access Key ID:

object_admin1

Access Key ID can be found here: https://console.aws.amazon.com/iam/home?#security_credential

Secret Access Key:

Secret Access Key can be found here: https://console.aws.amazon.com/iam/home?#security_credential

Figure A.15: Add account info

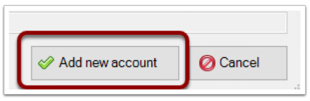
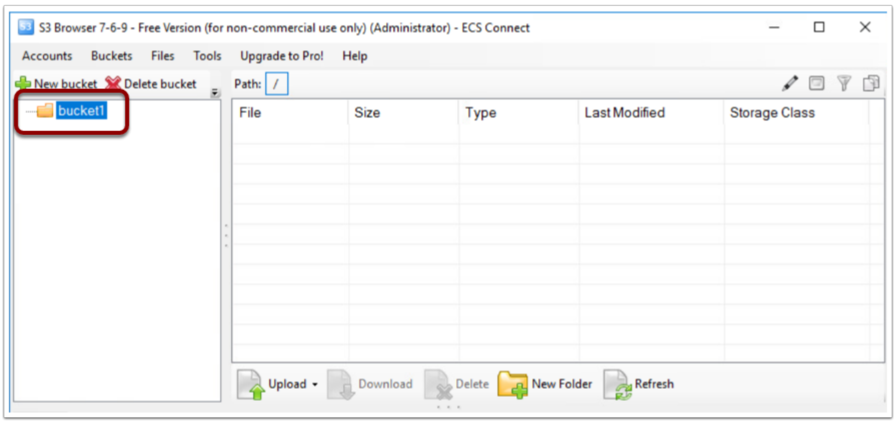


Figure A.16: Add account info



S3 Browser 7-6-9 - Free Version (for non-commercial use only) (Administrator) - ECS Connect

Figure A.17: Verify connection

5. Upload files to ECS through S3 browser.

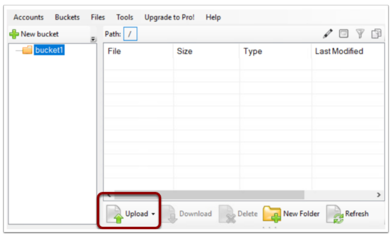
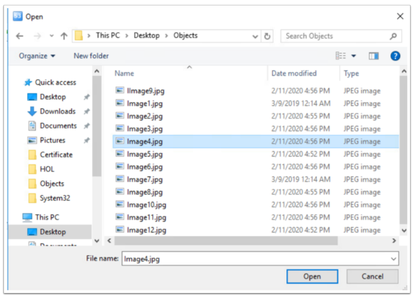


Figure A.18: Upload files to ECS through S3 browser

6. Navigate to objects folder on desktop,select all images.

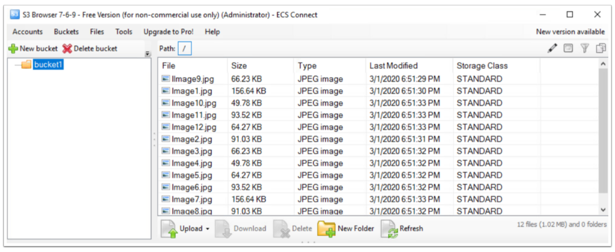


Name	Date modified	Type
IImage9.jpg	2/11/2020 4:56 PM	JPEG image
Image1.jpg	3/9/2019 12:14 AM	JPEG image
Image2.jpg	2/11/2020 4:55 PM	JPEG image
Image3.jpg	2/11/2020 4:56 PM	JPEG image
Image4.jpg	2/11/2020 4:56 PM	JPEG image
Image5.jpg	2/11/2020 4:52 PM	JPEG image
Image6.jpg	2/11/2020 4:56 PM	JPEG image
Image7.jpg	3/9/2019 12:14 AM	JPEG image
Image8.jpg	2/11/2020 4:55 PM	JPEG image
Image10.jpg	2/11/2020 4:56 PM	JPEG image
Image11.jpg	2/11/2020 4:56 PM	JPEG image
Image12.jpg	2/11/2020 4:52 PM	JPEG image

File name: Image4.jpg

Figure A.19: Select images

7. All of the images will be added to the ECS Storage through ECS Connection Manager.

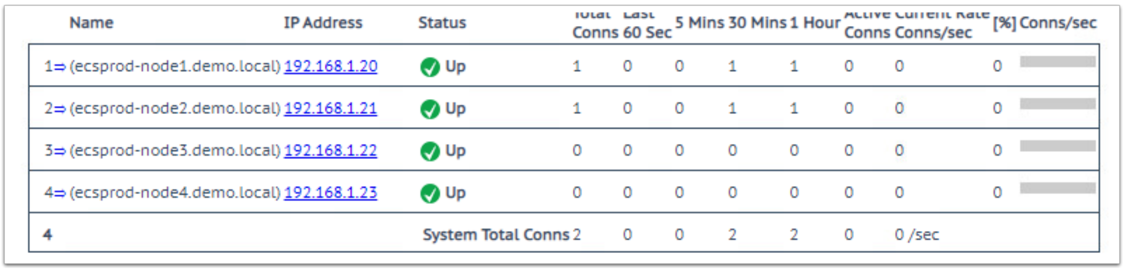


File	Size	Type	Last Modified	Storage Class
IImage9.jpg	66.23 KB	JPEG image	3/1/2020 6:51:29 PM	STANDARD
Image1.jpg	156.64 KB	JPEG image	3/1/2020 6:51:30 PM	STANDARD
Image10.jpg	49.78 KB	JPEG image	3/1/2020 6:51:33 PM	STANDARD
Image11.jpg	93.52 KB	JPEG image	3/1/2020 6:51:33 PM	STANDARD
Image12.jpg	64.27 KB	JPEG image	3/1/2020 6:51:33 PM	STANDARD
Image2.jpg	91.03 KB	JPEG image	3/1/2020 6:51:31 PM	STANDARD
Image3.jpg	66.23 KB	JPEG image	3/1/2020 6:51:32 PM	STANDARD
Image4.jpg	49.78 KB	JPEG image	3/1/2020 6:51:32 PM	STANDARD
Image5.jpg	64.27 KB	JPEG image	3/1/2020 6:51:32 PM	STANDARD
Image6.jpg	93.52 KB	JPEG image	3/1/2020 6:51:32 PM	STANDARD
Image7.jpg	156.64 KB	JPEG image	3/1/2020 6:51:33 PM	STANDARD
Image8.jpg	91.03 KB	JPEG image	3/1/2020 6:51:32 PM	STANDARD

12 files (1.02 MB) and 0 folders

Figure A.20: Images added to ECS

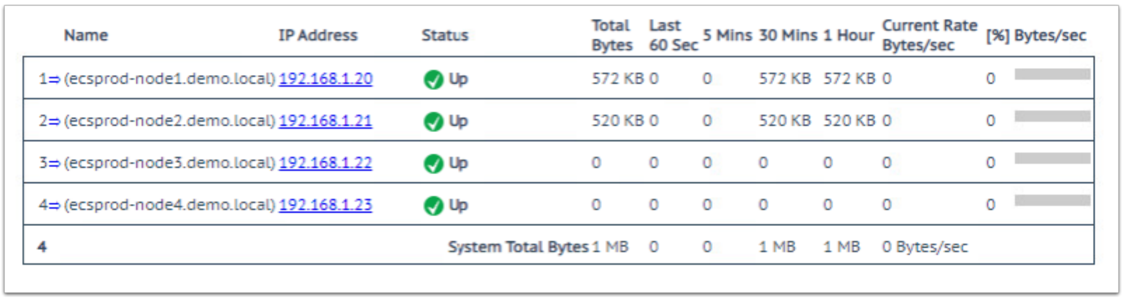
8. In the left-hand navigation click Statistics and then Real Time Statistics, choose real time servers.This will display the real time connections for each of the ECS Nodes.



Name	IP Address	Status	Total Conns	Last 60 Sec	5 Mins	30 Mins	1 Hour	Active Conns	Current Rate Conns/sec	[%] Conns/sec
1⇒ (ecsprod-node1.demo.local)	192.168.1.20	✓ Up	1	0	0	1	1	0	0	0
2⇒ (ecsprod-node2.demo.local)	192.168.1.21	✓ Up	1	0	0	1	1	0	0	0
3⇒ (ecsprod-node3.demo.local)	192.168.1.22	✓ Up	0	0	0	0	0	0	0	0
4⇒ (ecsprod-node4.demo.local)	192.168.1.23	✓ Up	0	0	0	0	0	0	0	0
4		System Total Conns	2	0	0	2	2	0	0 /sec	

Figure A.21: Images added to ECS

9. This will display the real time bytes for each of the ECS Nodes.



Name	IP Address	Status	Total Bytes	Last 60 Sec	5 Mins	30 Mins	1 Hour	Current Rate Bytes/sec	[%] Bytes/sec
1⇒ (ecsprod-node1.demo.local)	192.168.1.20	✓ Up	572 KB	0	0	572 KB	572 KB	0	0
2⇒ (ecsprod-node2.demo.local)	192.168.1.21	✓ Up	520 KB	0	0	520 KB	520 KB	0	0
3⇒ (ecsprod-node3.demo.local)	192.168.1.22	✓ Up	0	0	0	0	0	0	0
4⇒ (ecsprod-node4.demo.local)	192.168.1.23	✓ Up	0	0	0	0	0	0	0
4		System Total Bytes	1 MB	0	0	1 MB	1 MB	0 Bytes/sec	

Figure A.22: Images added to ECS

A Appendix B:Azure Storage

This appendix contains details about how to create an Azure Storage account, upload files on Blob, Table, File-share, and create a virtual machine on an un managed disk.

A.1 Creating Azure account on the Azure portal:

1. Login to Azure site using your Microsoft mail: www.azure.com
2. choose "Create resource" as in the below figure:

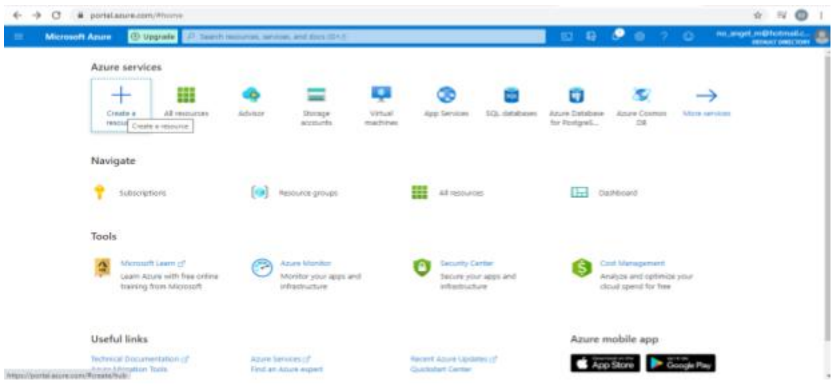


Figure A.1: Create a resource

3. Then choose from the column "Storage account"

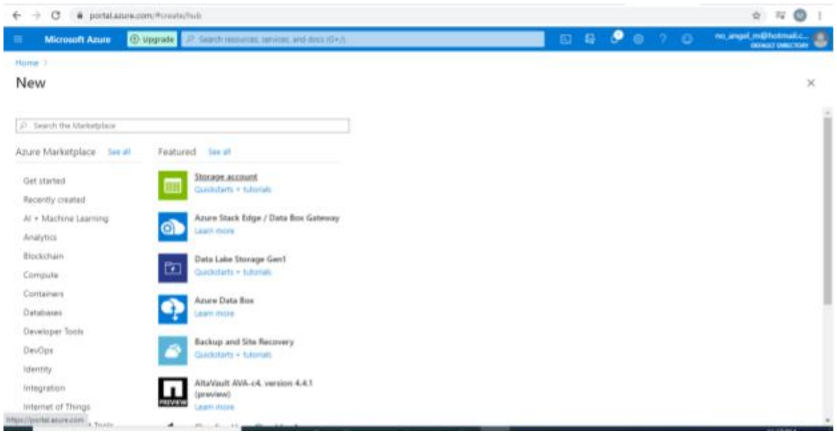
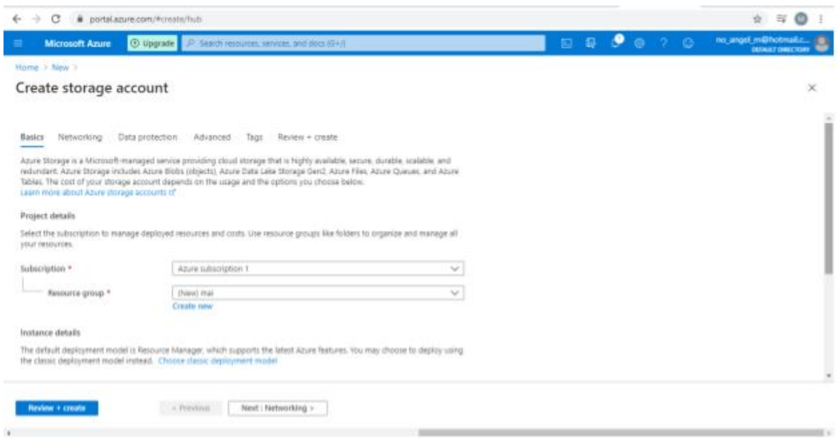


Figure A.2: Storage account

4. Create subscription and resource group as the below figure



Create storage account

Basics | Networking | Data protection | Advanced | Tags | Review + create

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.
Learn more about Azure storage accounts

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

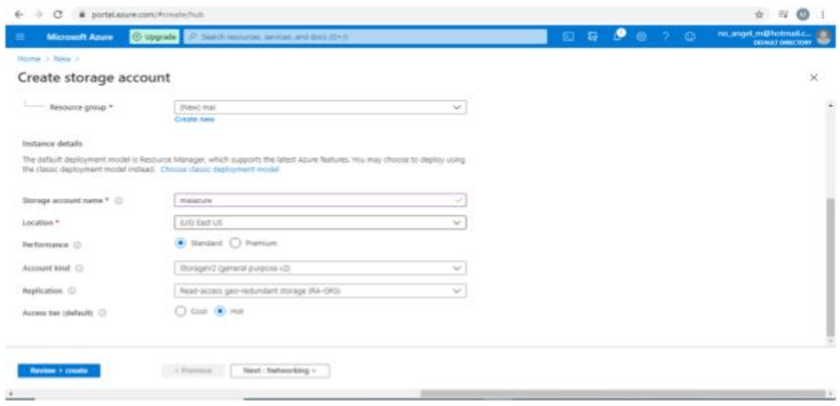
Subscription *: Azure subscription 1
Resource group *: (New) mai
Create new

Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Figure A.3: subscription and resource group

5. Creating a name for the storage account which must be a unique name within Azure.Also choosing the location where the data will be stored within it(by default it is East US.Choosing the Performance tier(the default option is standard which is based on HDD),Choose in the account type the default which is StorageV2,Replication is Read-access geo redundant storage(RA-GRS) and the last choice is the access tier which is by default hot.



Create storage account

Resource group *: (New) mai
Create new

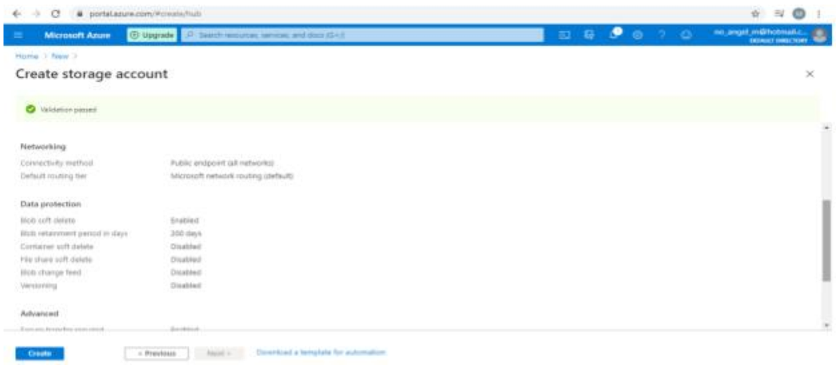
Instance details

The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Storage account name *: maiazure
Location *: (US) East US
Performance: [✓ Standard] Premium
Account kind: StorageV2 (general purpose v2)
Replication: Read-access geo-redundant storage (RA-GRS)
Access tier (default): Cool [✓ Hot]

Figure A.4: Creating Storage Account

6. Choose the default for all the next pages till the (review+create) page as the below figure:
7. By clicking create button the creation of the Storage account is done as the below figure:


portal.azure.com/#create/hub
Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Home > New >
Create
< Previous
Next >
Download a template for automation

Create storage account

Validation passed

Networking

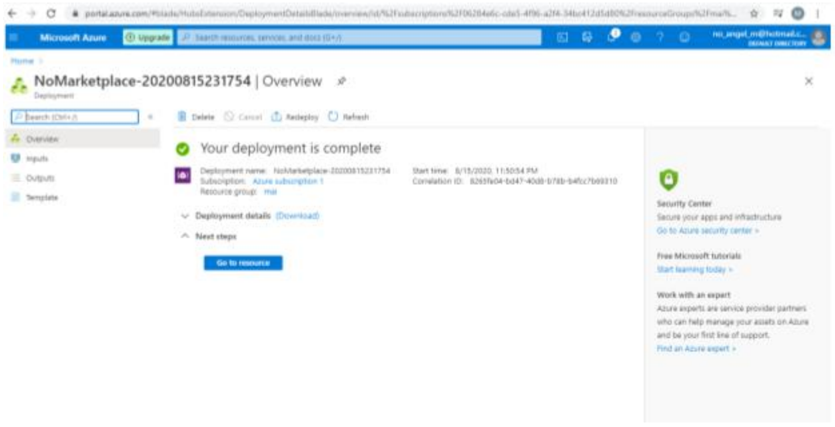
Connectivity method	Public endpoint (all networks)
Default routing tier	Microsoft network routing (default)

Data protection

Blob soft delete	Enabled
Blob retainment period in days	300 days
Container soft delete	Disabled
File share soft delete	Disabled
Blob change feed	Disabled
Versioning	Disabled

Advanced

Figure A.5: Review and Create Storage Account



NoMarketplace-20200815231754 | Overview

Deployment

Your deployment is complete

Deployment name: NoMarketplace-20200815231754
Subscription: Azure subscription 1
Resource group: mai

Start time: 8/15/2020, 11:50:54 PM
Correlation ID: 8265fe04-bd47-40d8-b78b-b4fcc7b69310

Deployment details (Download)

Next steps

Security Center
Secure your apps and infrastructure
Go to Azure security center >

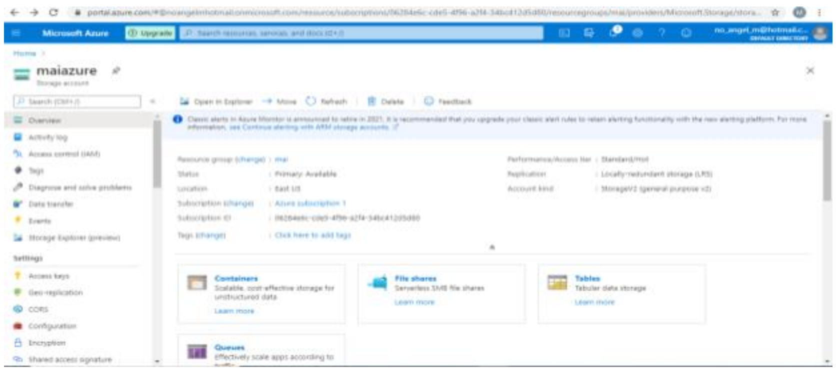
Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

Figure A.6: verify the creation of Storage Account

A.2 Create Blob service and upload a file to it

1. Create the container where the blob is stored on it, Choose container from the overview page as the figure below:



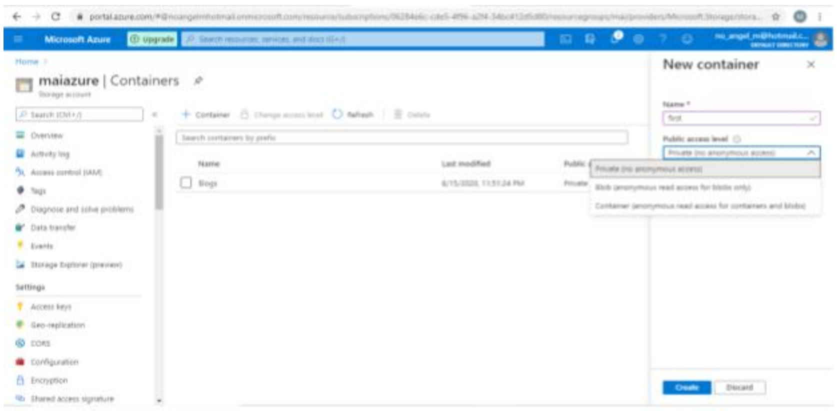
Classic alerts in Azure Monitor is announced to retire in 2021. It is recommended that you upgrade your classic alert rules to retain alerting functionality with the new alerting platform. For more information, see Continue alerting with ARM storage accounts.

Resource group (change) : mai
Status : Primary: Available
Location : East US
Subscription (change) : Azure subscription 1
Subscription ID : 06284e6c-cde5-4f96-a2f4-54bc412d5d80
Tags (change) : Click here to add tags
Performance/Access tier : Standard/Hot
Replication : Locally-redundant storage (LRS)
Account kind : StorageV2 (general purpose v2)

Containers: Scalable, cost-effective storage for unstructured data. Learn more
File shares: Serverless SMB file shares. Learn more
Tables: Tabular data storage. Learn more

Figure A.7: choose the container service

2. Create new container by clicking on “+ container” bottom, then name this container as the figure below:

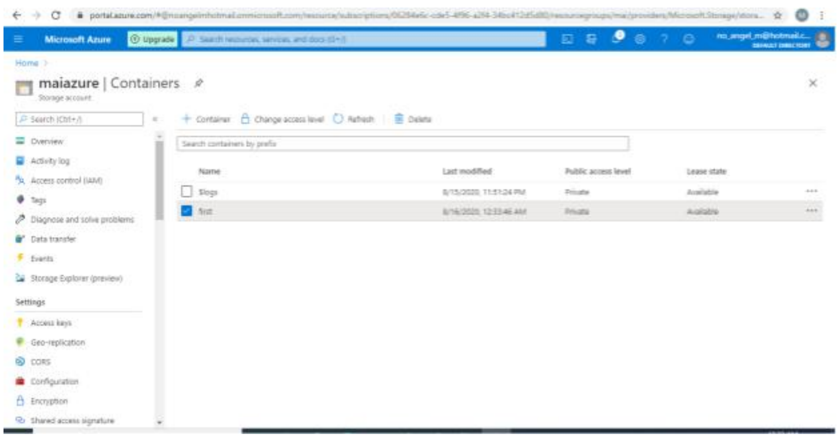


Blogs 6/15/2020, 11:51:24 PM Private

New container
Name * : first
Public access level : Private (no anonymous access)
Private (no anonymous access)
Blob (anonymous read access for blobs only)
Container (anonymous read access for containers and blobs)

Figure A.8: Creating of a new container

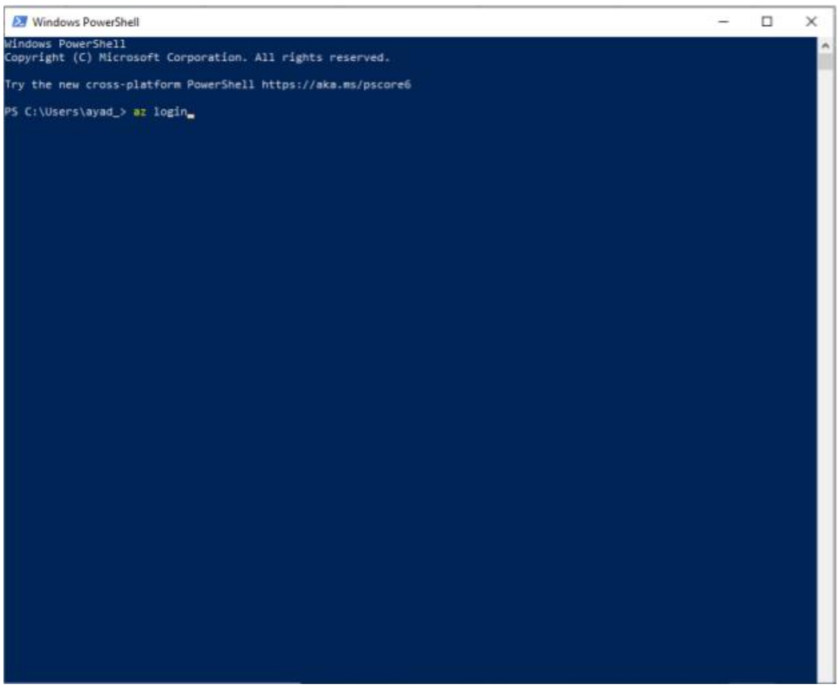
3. The container which we name”first” is created now as shown in the figure below.



Name	Last modified	Public access level	Lease state
$logs	8/15/2020, 11:51:24 PM	Private	Available
first	8/16/2020, 12:33:46 AM	Private	Available

Figure A.9: The end of container creation

4. Uploading a file to the blob using the power shell and the Az command butt first we need to download Azure CLI and then use the Windows PowerShell to login to the Azure portal using the command "az login" after this a new window asking for the Azure username and password as shown in the below figure:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ayad_> az login
```

Figure A.10: Login to the Azure portal using PowerShell

5. Uploading a video file with size 1G using this command.

```
az storage blob upload --account-name mystorageaccount

--account-key 0000-0000 --container-name mycontainer --file /path/to/file
--name myblob
```

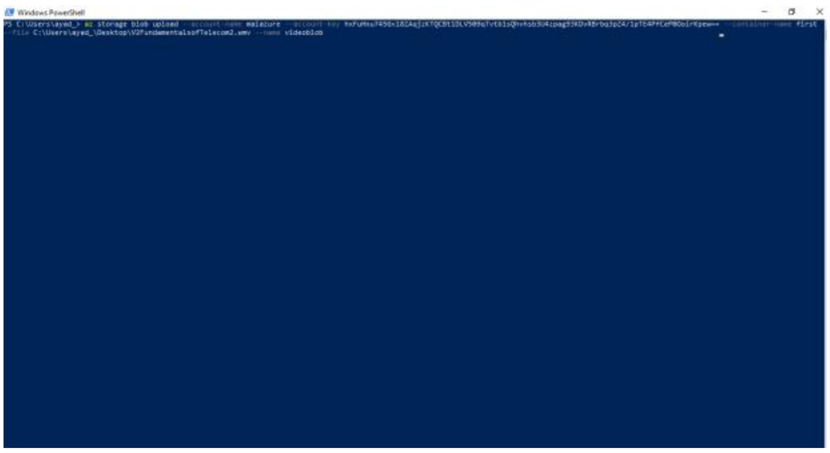
In our scenario :

- the account-name is ¡maiazure¿
- account key is $hxFuHxu745Gvl8ZAqjzKTQCBt1DLV509qTvtb1sQhvhsb3$
)
- Container-name is ¡first¿
- File path
- is$C : Users - ayad_Desktop - V2FundamentalsofTelecom2.wmv$
- Name is “videoblob”

So the command will be as the following:

```
az storage blob upload --account-name maiazure --account-key
hxFuHxu745Gvl8ZAqjzKTQCBt1DLV509qTvtb1sQhvhsb3U4zpag93KDvRBrbq3pZ4/1pTE4PfCeP0OoirKpew
== --container-name first --file C:\Users\ayad_\Desktop\V2FundamentalsofTelecom2.wmv --name
videoblob
```

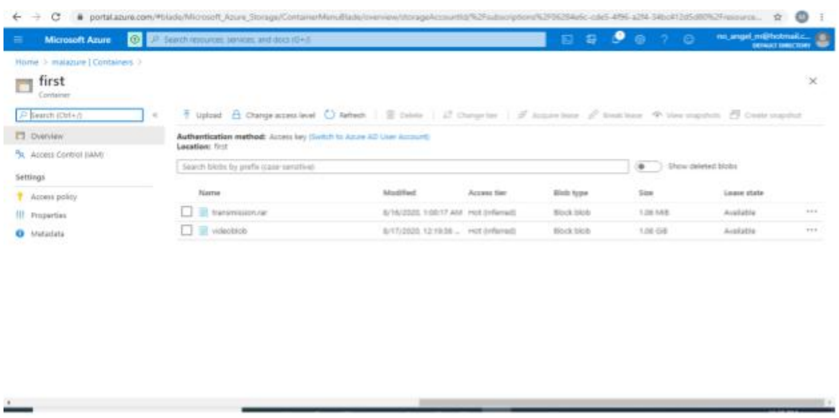
6. The figure below shows uploading the file to the blob.



```
PS C:\Users\ayad_> az storage blob upload --account-name maiazure --account-key hxFuHxu745Gvl8ZAqjzKTQCBt1DLV509qTvtb1sQhvhsb3U4zpag93KDvRBrbq3pZ4/1pTE4PfCeP0OoirKpew== --container-name first
--file C:\Users\ayad_\Desktop\V2FundamentalsofTelecom2.wmv --name videoblob
```

Figure A.11: uploading a file to Blob 1

7. When accessing the Azure portal we will find that the file is existing now as shown in the below figure:
8. On Azure portal there are many tools help to analysis azure service(Blob,Queue,Table,
   and disk) as shown in the figure below:
9. Click “Metric” in the left column, then click “add metric” and choose the metric you want to help you in your analysis. The figure below shows the capacity.
10. We can show many metrics in one chart, the figure below shows the Availability, Ingress, Egress, Success E2E latency, Success Server latency and Transactions metrics in one chart.
11. There are another way for analysis from the overview page, you can choose the account(container or file or table or queue) and then four charts will be displayed four four matrices which are Total Ingress, Total Egress, Average Latency, Request Break down as shown in the figure below.



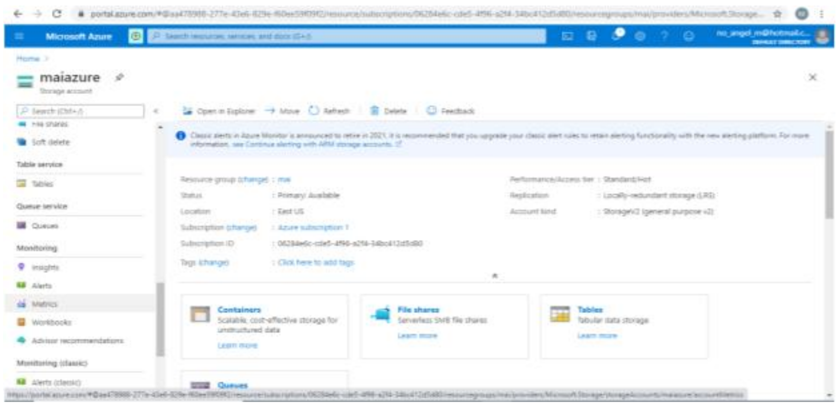
Home > maiazure | Containers >

first
Container

Authentication method: Access key (Switch to Azure AD User Account)
Location: first

Name	Modified	Access tier	Blob type	Size	Lease state
transmission.rar	8/16/2020, 1:00:17 AM	Hot (Inferred)	Block blob	1.08 MiB	Available
videoblob	8/17/2020, 12:19:38 ...	Hot (Inferred)	Block blob	1.08 GiB	Available

Figure A.12: verification of the uploaded



maiazure
Storage account

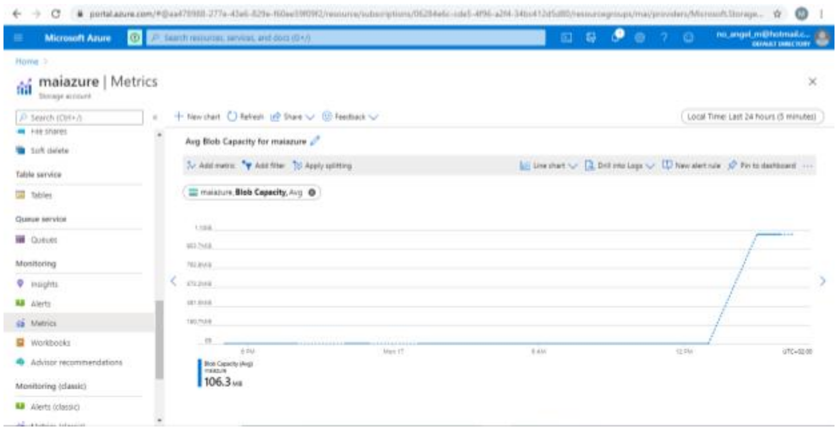
Classic alerts in Azure Monitor is announced to retire in 2021. It is recommended that you upgrade your classic alert rules to retain alerting functionality with the new alerting platform. For more information, see Continue alerting with ARM storage accounts.

Resource group (change) : mai
Status : Primary: Available
Location : East US
Subscription (change) : Azure subscription 1
Subscription ID : 06284e6c-cde5-4f96-a2f4-34bc412d5d80
Tags (change) : Click here to add tags

Performance/Access tier : Standard/Hot
Replication : Locally-redundant storage (LRS)
Account kind : StorageV2 (general purpose v2)

Containers — Scalable, cost-effective storage for unstructured data
File shares — Serverless SMB file shares
Tables — Tabular data storage

Figure A.13: choosing the metrics to analysis blob service



maiazure | Metrics
Storage account

Avg Blob Capacity for maiazure

maiazure, Blob Capacity, Avg

Local Time: Last 24 hours (5 minutes)

Blob Capacity (Avg)
maiazure
106.3 MB

Figure A.14: Analysis based on the capacity

12. Regarding the price metric ,Azure offers a site https://azure.microsoft.com/en-us/pricing/calculator/ that gives you the cost according to some feature which are region, account type, performance tier, storage account type, redundancy type, access tier and capacity taken in consideration that any change in these factors affect the cost.
The figure below shows the cost of the account that we created according to the features we chose

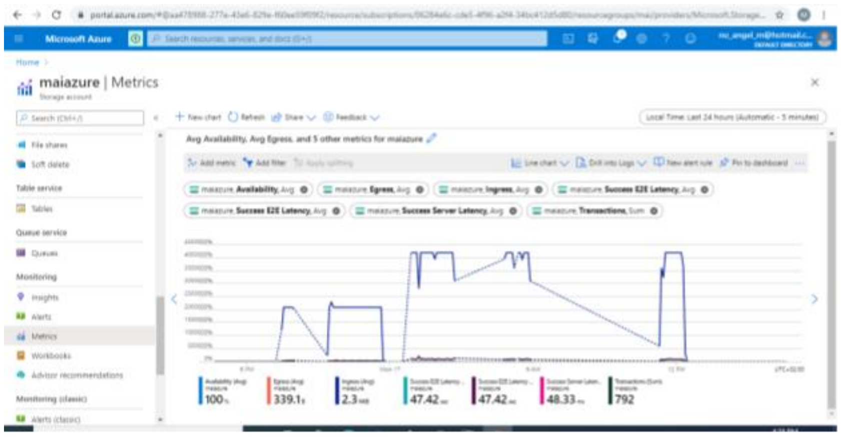


Figure A.15: Availability, Ingress, Egress, Success E2E latency, Success Server latency and Transactions metrics

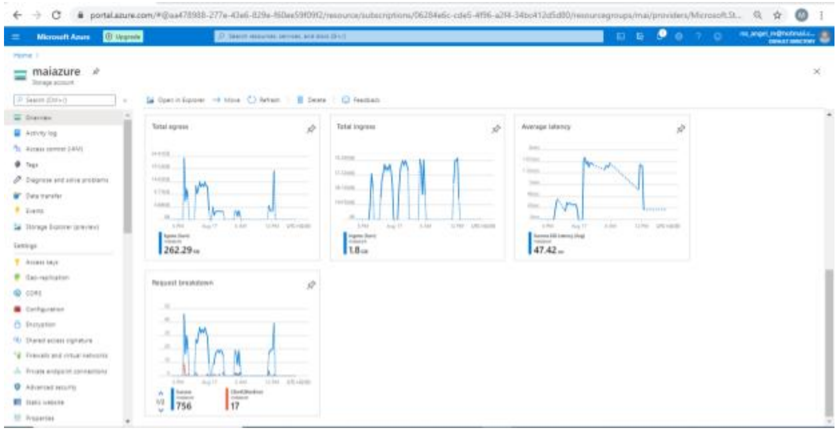
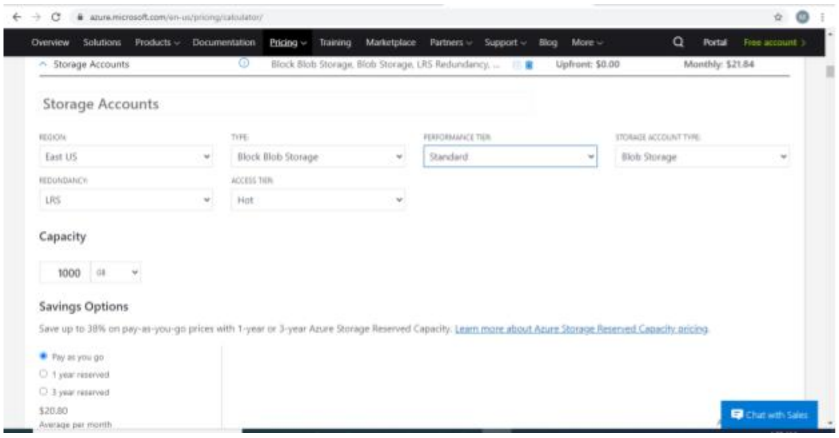


Figure A.16: available metrics in the overview page


Storage Accounts — Block Blob Storage, Blob Storage, LRS Redundancy, ... Upfront: $0.00 Monthly: $21.84

Storage Accounts

REGION: East US
TYPE: Block Blob Storage
PERFORMANCE TIER: Standard
STORAGE ACCOUNT TYPE: Blob Storage
REDUNDANCY: LRS
ACCESS TIER: Hot

Capacity

1000 GB

Savings Options

Save up to 38% on pay-as-you-go prices with 1-year or 3-year Azure Storage Reserved Capacity. Learn more about Azure Storage Reserved Capacity pricing.

[✓ Pay as you go]
1 year reserved
3 year reserved

$20.80
Average per month

Figure A.17: The cost of the account created

A.3 Create Queue service and add messages to it

1. After accessing Azure portal and creating storage account, we will create queue service.First access the Queue service from the overview page as shown in the figure below

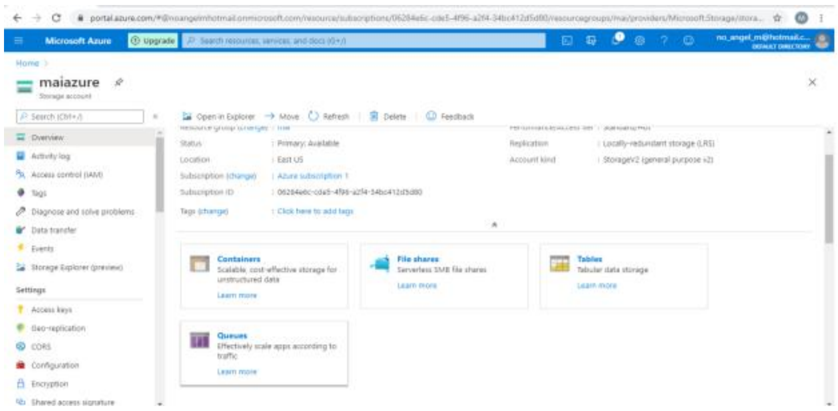


Figure A.18: Accessing the queue service

2. Add Queue, by clicking on "+ Queue" as shown on the figure below:

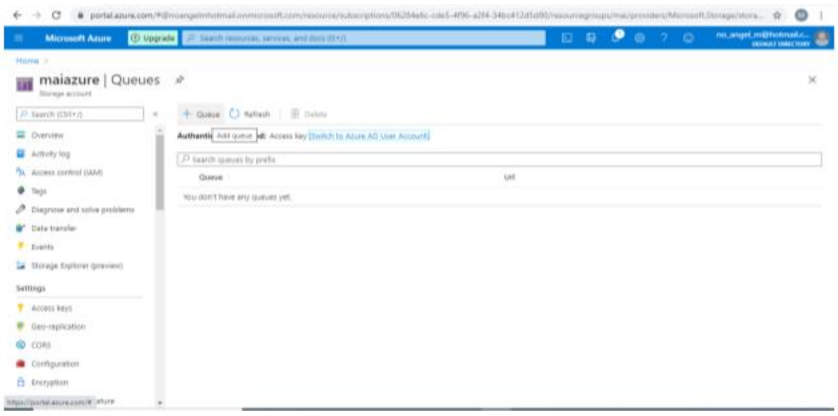


Figure A.19: Add new Queue

3. Then give name to this Queue as show in the figure below:
4. Now we create a Queue and named it "mai".
5. To add a message to the Queue, we first access it and then click"+Add message" as shown in the figure below.
6. After adding a message to the Queue,begin to write the text you want inside this message and choose the expiration date for this message(by default it is 7days),you can too choose to be expired as shown in the figure below:
7. After adding the messages, you can delete one of these message by clicking "dequeue message", the first message will be deleted first as shown in the figure below.
8. Also we can clear the whole Queue by clicking on"Clear Queue"

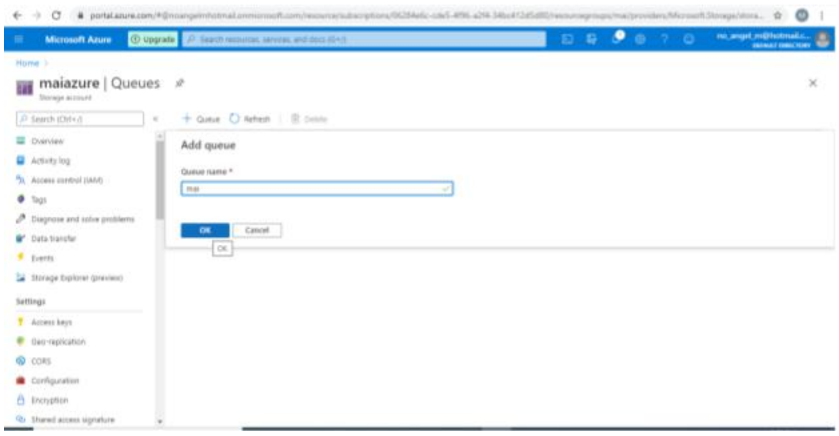


Figure A.20: Give a name to the Queue

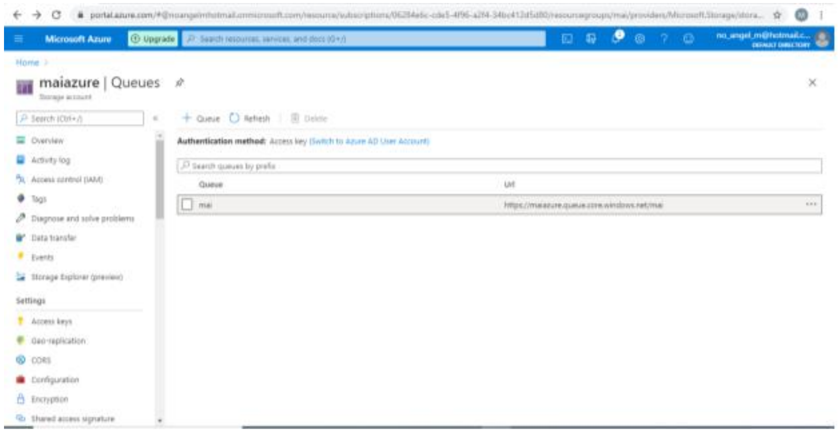


Figure A.21: End the step of creating the Queue and giving it a name

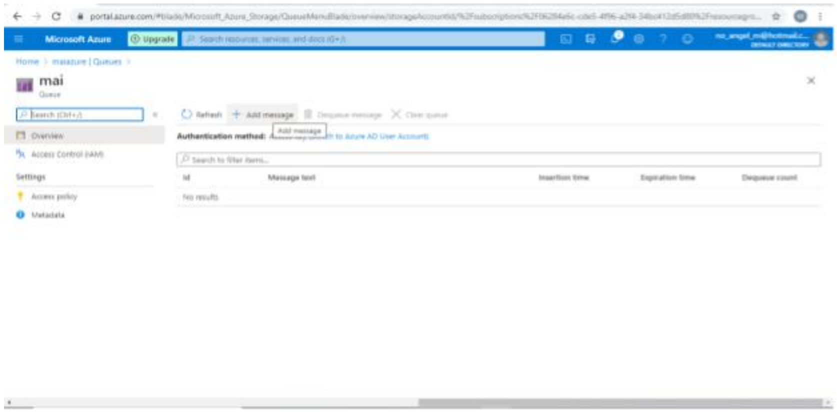
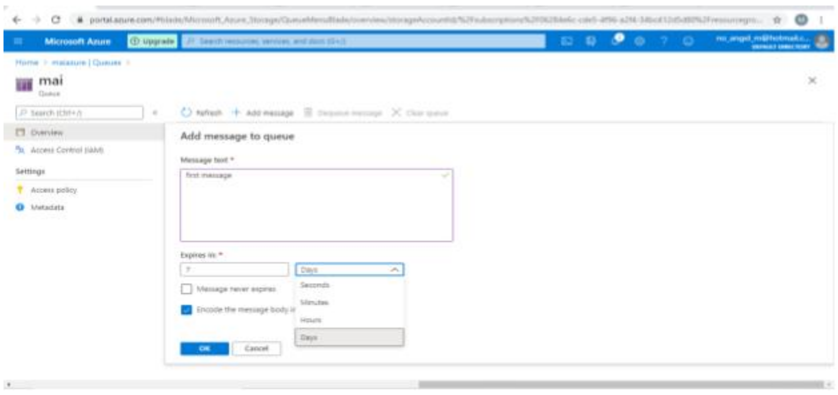


Figure A.22: Adding a message to the Queue

9. You can analyze the Queue service by clicking “Metric” in the left column,then click “add metric” and choose the metric you want to help you in your analysis. The figure below shows the capacity
10. We can show many metrics in one chart, the figure below shows the Availability,Ingress,Egress,Success E2E latency,Success Server latency and Transactions metrics in one chart.



Add message to queue

Message text *

first message

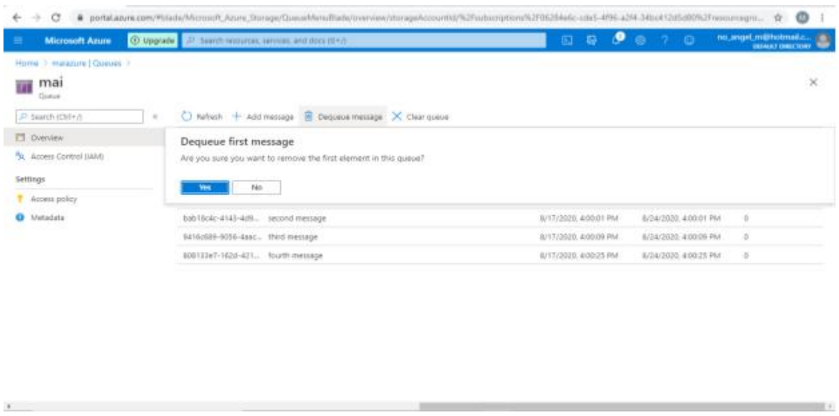
Expires in: *

7 Days

Message never expires

Encode the message body i

Figure A.23: Adding a text to the message

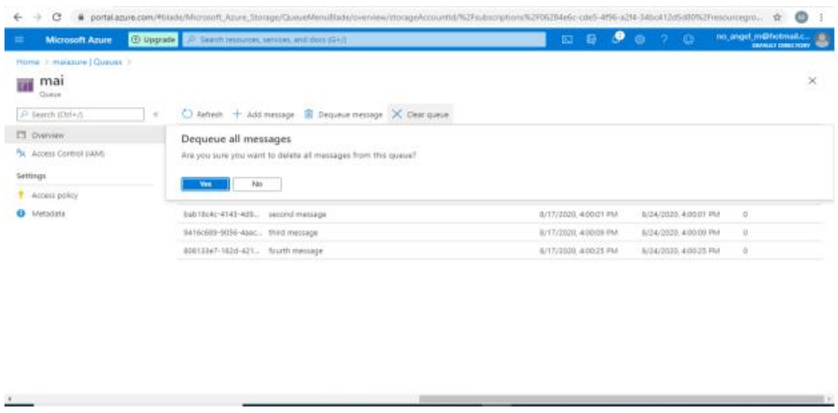


Dequeue first message

Are you sure you want to remove the first element in this queue?

bab18c4c-4143-4d9...	second message	8/17/2020, 4:00:01 PM	8/24/2020, 4:00:01 PM	0
9416c689-9056-4aac...	third message	8/17/2020, 4:00:09 PM	8/24/2020, 4:00:09 PM	0
808133e7-162d-421...	fourth message	8/17/2020, 4:00:25 PM	8/24/2020, 4:00:25 PM	0

Figure A.24: Deleting a message from the Queue



Dequeue all messages

Are you sure you want to delete all messages from this queue?

bab18c4c-4143-4d9...	second message	8/17/2020, 4:00:01 PM	8/24/2020, 4:00:01 PM	0
9416c689-9056-4aac...	third message	8/17/2020, 4:00:09 PM	8/24/2020, 4:00:09 PM	0
808133e7-162d-421...	fourth message	8/17/2020, 4:00:25 PM	8/24/2020, 4:00:25 PM	0

Figure A.25: Clearing the Queue 1

11. There are another way for analysis from the overview page,you can choose the account(container or file or table or queue) and then four charts will be displayed four four matrices which are Total Ingress,Total Egress,Average Latency,Request Break down as shown in the figure below.
12. Regarding the price matric ,Azure offers a site https://azure.microsoft.com/en-us/pricing/calculator/ that gives you the cost according to some feature which are region,account type,performance tier,storage account type,redundancy type,access tier and capacity taken in consideration that any change in these

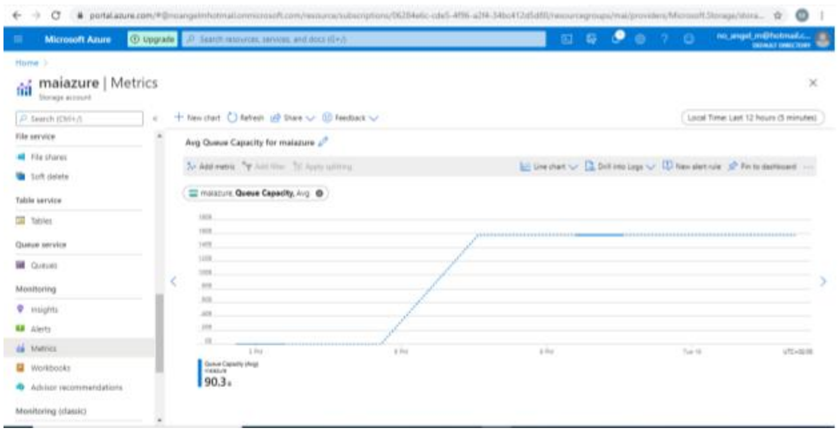
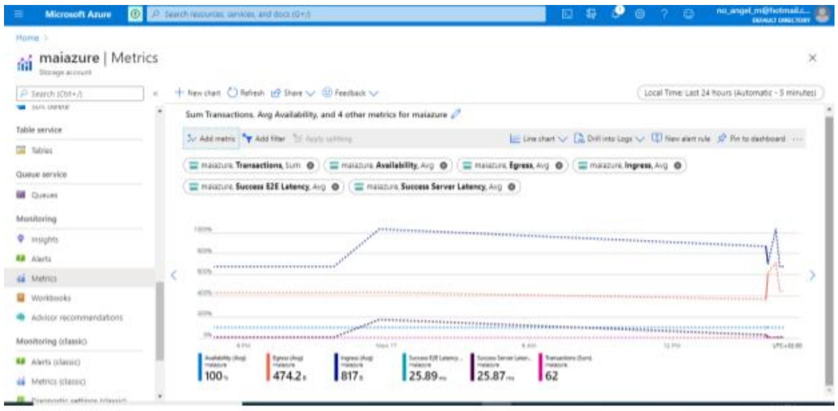


Figure A.26: Analysis based on the capacity matric



maiazure | Metrics
Sum Transactions, Avg Availability, and 4 other metrics for maiazure
100
474.2
817
25.89
25.87
62

Figure A.27: Availability,Ingress,Egress,Success E2E latency,Success Server latency and Transactions metrics

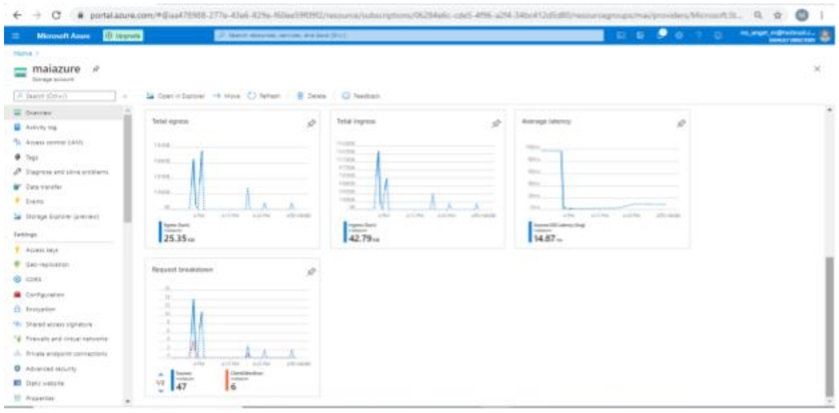
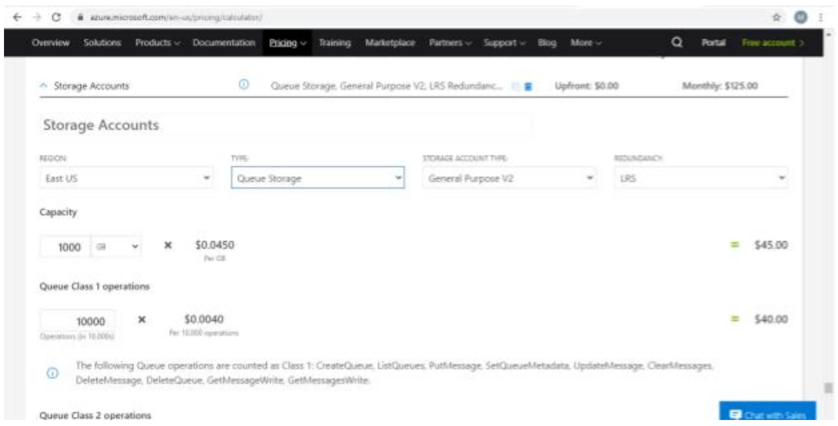


Figure A.28: available metrics in the overview page

factors affect the cost The figure below shows the cost of the account that we created according to the features we chose



Storage Accounts — Queue Storage, General Purpose V2, LRS Redundanc... — Upfront: $0.00 — Monthly: $125.00

Storage Accounts

REGION:	TYPE:	STORAGE ACCOUNT TYPE:	REDUNDANCY:
East US	Queue Storage	General Purpose V2	LRS

Capacity

1000 GB × $0.0450 Per GB = $45.00

Queue Class 1 operations

10000 Operations (in 10,000s) × $0.0040 Per 10,000 operations = $40.00

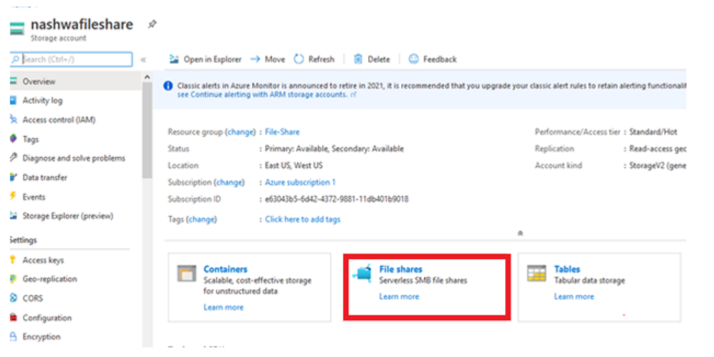
The following Queue operations are counted as Class 1: CreateQueue, ListQueues, PutMessage, SetQueueMetadata, UpdateMessage, ClearMessages, DeleteMessage, DeleteQueue, GetMessageWrite, GetMessagesWrite.

Queue Class 2 operations

Figure A.29: calculating the cost

A.4 Creating a file share

1. In order to create a file share or (unmanageddisk) you have to create a storage account first please refer to section.(4.1 Creating Azure account on the Azure Portal).



nashwafileshare
Storage account

Classic alerts in Azure Monitor is announced to retire in 2021, it is recommended that you upgrade your classic alert rules to retain alerting functionali
see Continue alerting with ARM storage accounts.

Resource group (change)	: File-Share
Status	: Primary: Available, Secondary: Available
Location	: East US, West US
Subscription (change)	: Azure subscription 1
Subscription ID	: e63043b5-6d42-4372-9881-11db401b9018
Tags (change)	: Click here to add tags
Performance/Access tier	: Standard/Hot
Replication	: Read-access geo
Account kind	: StorageV2 (gene

Containers Scalable, cost-effective storage for unstructured data. Learn more

File shares Serverless SMB file shares. Learn more

Tables Tabular data storage. Learn more

Figure 4.30 Storage Account Overview Activat

Figure A.30: Storage Account Overview

2. Click on file share

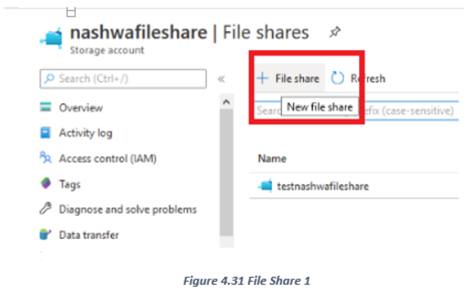
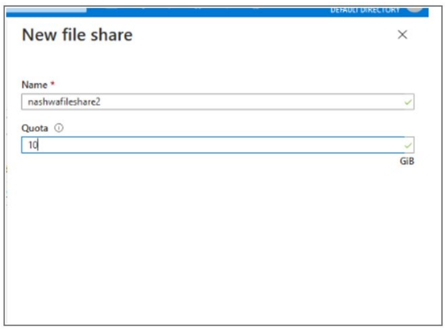


Figure A.31: File share1



Name *
nashwafileshare2
Quota
10
GiB

Figure A.32: File share2

A.5 Connecting to a file share

3. Log in to your azure portal and navigate to the share.

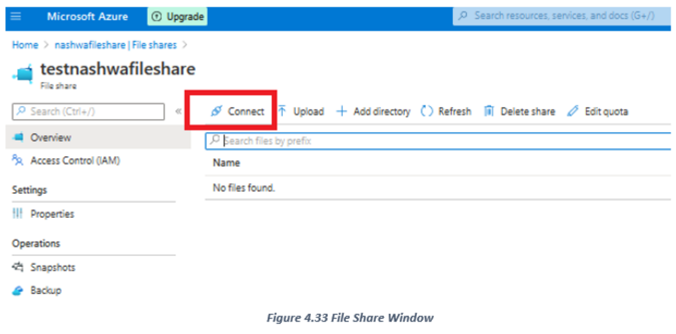
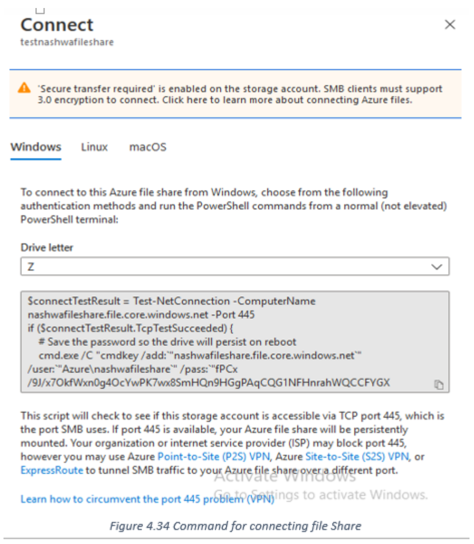


Figure A.33: File Share Window

4. Click connect and Copy the script that is written in the gray box .


Connect
testnashwafileshare

'Secure transfer required' is enabled on the storage account. SMB clients must support 3.0 encryption to connect. Click here to learn more about connecting Azure files.

Windows Linux macOS

To connect to this Azure file share from Windows, choose from the following authentication methods and run the PowerShell commands from a normal (not elevated) PowerShell terminal:

Drive letter: Z

```
$connectTestResult = Test-NetConnection -ComputerName
nashwafileshare.file.core.windows.net -Port 445
if ($connectTestResult.TcpTestSucceeded) {
   # Save the password so the drive will persist on reboot
   cmd.exe /C "cmdkey /add:`"nashwafileshare.file.core.windows.net`"
/user:`"Azure\nashwafileshare`" /pass:`"fPCx
/9J/x7OkfWxn0g4OcYwPK7wx8SmHQn9HGgPAqCQG1NFHnrahWQCCFYGX
```

This script will check to see if this storage account is accessible via TCP port 445, which is the port SMB uses. If port 445 is available, your Azure file share will be persistently mounted. Your organization or internet service provider (ISP) may block port 445, however you may use Azure Point-to-Site (P2S) VPN, Azure Site-to-Site (S2S) VPN, or ExpressRoute to tunnel SMB traffic to your Azure file share over a different port.

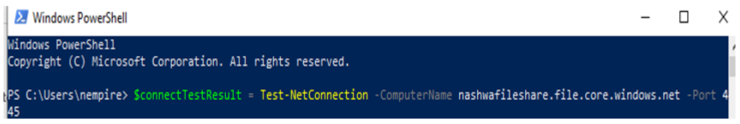
Learn how to circumvent the port 445 problem (VPN)

Activate Windows
Go to Settings to activate Windows.

Figure 4.34 Command for connecting file Share

Figure A.34: Command for connecting file Share

5. Open windows power shell on your computer,Paste the command you typed in PowerShell CMD

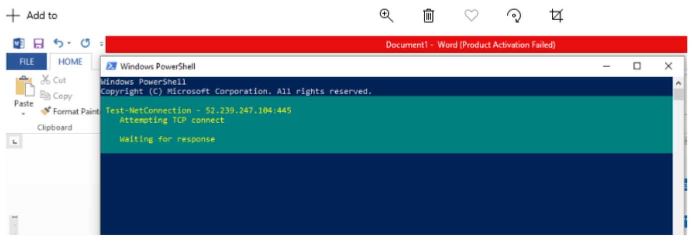


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\nempire> $connectTestResult = Test-NetConnection -ComputerName nashwafileshare.file.core.windows.net -Port 4
45
```

Figure 4.35 connecting file using power Shell

Figure A.35: connecting file using power Shell

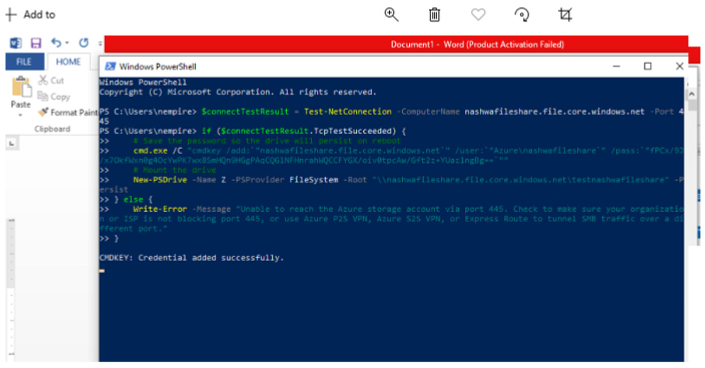
6. In The figure below, the pc is trying to connect share through port 445



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Test-NetConnection - 52.239.247.104:445
   Attempting TCP connect

   Waiting for response
```

Figure 4.36 connecting file using power shell running

Figure A.36: connecting file using power shell running

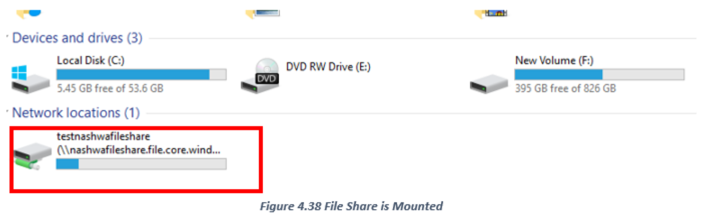


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\nempire> $connectTestResult = Test-NetConnection -ComputerName nashwafileshare.file.core.windows.net -Port 4
45
PS C:\Users\nempire> if ($connectTestResult.TcpTestSucceeded) {
>>     # Save the password so the drive will persist on reboot
>>     cmd.exe /C "cmdkey /add:`"nashwafileshare.file.core.windows.net`" /user:`"Azure\nashwafileshare`" /pass:`"fPCx/9J
/x7OkfWxn0g4OcYwPK7wx8SmHQn9HGgPAqCQG1NFHnrahWQCCFYGX/oiv0tpcAw/Gft2z+YUaz1ngBg==`""
>>     # Mount the drive
>>     New-PSDrive -Name Z -PSProvider FileSystem -Root "\\nashwafileshare.file.core.windows.net\testnashwafileshare" -P
ersist
>> } else {
>>     Write-Error -Message "Unable to reach the Azure storage account via port 445. Check to make sure your organizatio
n or ISP is not blocking port 445, or use Azure P2S VPN, Azure S2S VPN, or Express Route to tunnel SMB traffic over a di
fferent port."
>> }

CMDKEY: Credential added successfully.
```

Figure 4.37 connecting file using power shell END

Figure A.37: connecting file using power shell END

7. open file Explorer, the “nashwatestfileshare” will be mounted as shown below.



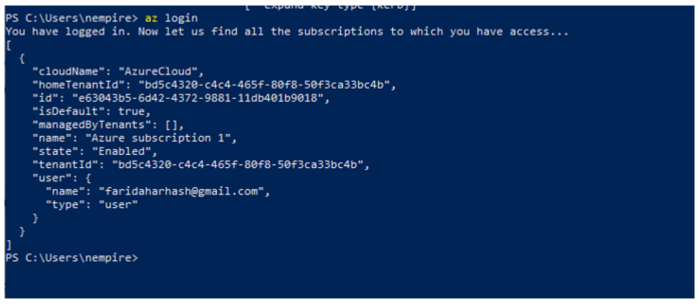
Devices and drives (3)
Local Disk (C:)
5.45 GB free of 53.6 GB
DVD RW Drive (E:)
New Volume (F:)
395 GB free of 826 GB
Network locations (1)
testnashwafileshare
(\\nashwafileshare.file.core.wind...

Figure 4.38 File Share is Mounted

Figure A.38: File Share is Mounted

A.6 Uploading files to file share

1. Login to Azure through cli, you will be prompted by a browser page asking you for your email and password, once you gave your credential, you will be redirected to the PowerShell again

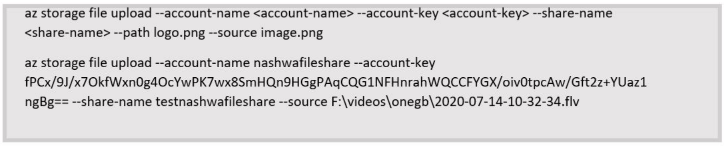


```
PS C:\Users\nempire> az login
You have logged in. Now let us find all the subscriptions to which you have access...
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "bd5c4320-c4c4-465f-80f8-50f3ca33bc4b",
    "id": "e63043b5-6d42-4372-9881-11db401b9018",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure subscription 1",
    "state": "Enabled",
    "tenantId": "bd5c4320-c4c4-465f-80f8-50f3ca33bc4b",
    "user": {
      "name": "faridaharhash@gmail.com",
      "type": "user"
    }
  }
]
PS C:\Users\nempire>
```

Figure 4.39 Login to Azure using CLI

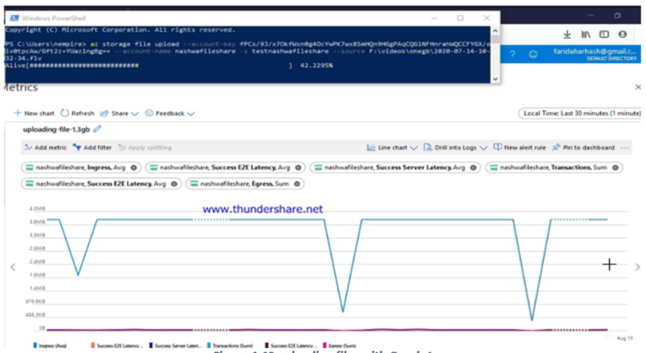
Figure A.39: Login to Azure using CLI

2. Type the following command in order to upload a video file to your Azure File share.



```
az storage file upload --account-name <account-name> --account-key <account-key> --share-name
<share-name> --path logo.png --source image.png

az storage file upload --account-name nashwafileshare --account-key
fPCx/9J/x7OkfWxn0g4OcYwPK7wx8SmHQn9HGgPAqCQG1NFHnrahWQCCFYGX/oiv0tpcAw/Gft2z+YUaz1
ngBg== --share-name testnashwafileshare --source F:\videos\onegb\2020-07-14-10-32-34.flv
```

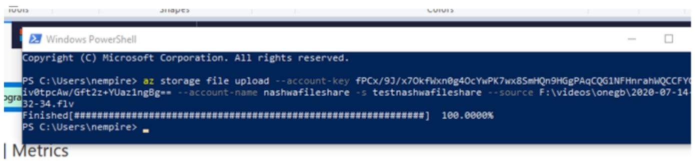
- Wait until upload reaches 100 percent, you will notice while data is transferred to cloud the ingress rate is increasing.
- Note that you can export the values of this graph in excel sheet as shown below.



```
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\nempire> az storage file upload --account-key fPCx/9J/x7OkfWxn0g4OcYwPK7wx8SmHQn9HGgPAqCQG1NFHnrahWQCCFYGX/o
iv0tpcAw/Gft2z+YUaz1ngBg== --account-name nashwafileshare -s testnashwafileshare --source F:\videos\onegb\2020-07-14-10-
32-34.flv
Alive[############################                              ]  42.2295%
```

Figure 4.40 uploading files with Graph 1

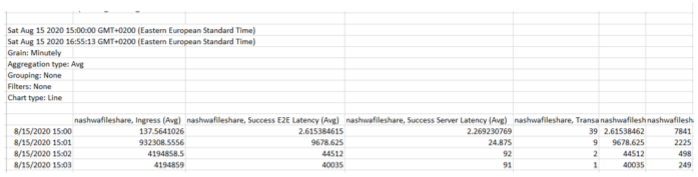
Figure A.40: uploading files with Graph 1



```
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\nempire> az storage file upload --account-key fPCx/9J/x7OkfWxn0g4OcYwPK7wx8SmHQn9HGgPAqCQG1NFHnrahWQCCFY(
iv0tpcAw/Gft2z+YUaz1ngBg== --account-name nashwafileshare -s testnashwafileshare --source F:\videos\onegb\2020-07-14-
32-34.flv
Finished[##############################################################]  100.0000%
PS C:\Users\nempire> _
```

Figure 4.41 uploading file is completed.

Figure A.41: uploading file is completed.



Sat Aug 15 2020 15:00:00 GMT+0200 (Eastern European Standard Time)
Sat Aug 15 2020 16:55:13 GMT+0200 (Eastern European Standard Time)
Grain: Minutely
Aggregation type: Avg
Grouping: None
Filters: None
Chart type: Line

	nashwafileshare, Ingress (Avg)	nashwafileshare, Success E2E Latency (Avg)	nashwafileshare, Success Server Latency (Avg)	nashwafileshare, Transa	nashwafilesh	nashwafilesh
8/15/2020 15:00	137.5641026	2.615384615	2.269230769	39	2.61538462	7841
8/15/2020 15:01	932308.5556	9678.625	24.875	9	9678.625	2225
8/15/2020 15:02	4194858.5	44512	92	2	44512	498
8/15/2020 15:03	4194859	40035	91	1	40035	249

Figure 4.43 Metrics in Excel sheet

Figure A.42: Metrics in Excel sheet

A.7 Azure Cosmos DB

1. login to azure portal and navigate to "Azure Cosmos DB"

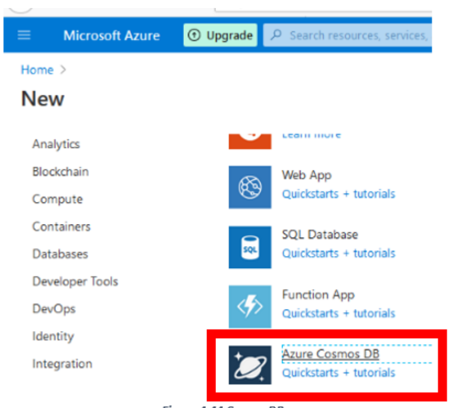
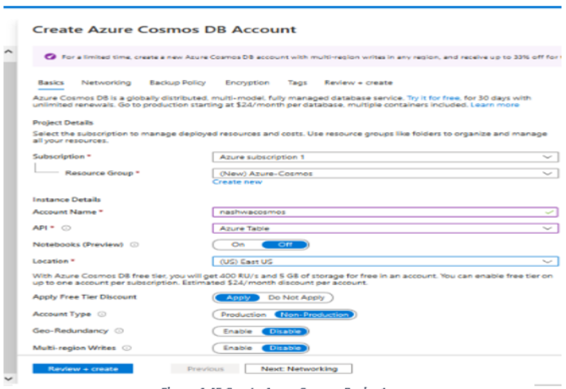


Figure 4.44 CosmosDB

Figure A.43: Cosmos DB

2. By clicking "Azure Cosmos DB" a wizard will be prompted to you asking you to fill The resource group (you can think of resource group a folder that combines objects that related to each other for example "Virtual machines that belongs to the web project")In the API field select "Azure Table" Here you can select the location you need and accept the default values in the upcoming fields.



Create Azure Cosmos DB Account

For a limited time, create a new Azure Cosmos DB account with multi-region writes in any region, and receive up to 33% off for

Basics Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a globally distributed, multi-model, fully managed database service. Try it for free, for 30 days with unlimited renewals. Go to production starting at $24/month per database, multiple containers included. Learn more

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Azure subscription 1
Resource Group *	(New) Azure-Cosmos Create new

Instance Details

Account Name *	nashwacosmos
API *	Azure Table
Notebooks (Preview)	On [✓ Off]
Location *	(US) East US

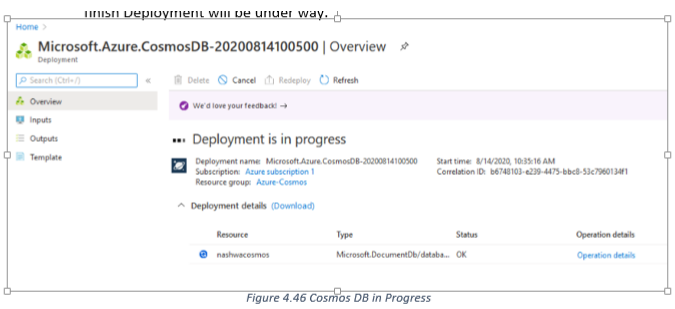
With Azure Cosmos DB free tier, you will get 400 RU/s and 5 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated $24/month discount per account.

Apply Free Tier Discount	[✓ Apply] Do Not Apply
Account Type	Production [✓ Non-Production]
Geo-Redundancy	Enable [✓ Disable]
Multi-region Writes	Enable [✓ Disable]

Figure 4.45 Create Azure Cosmos Basics 1

Figure A.44: Create Azure Cosmos Basics 1

3. Just click next to the following configuration parts accepting the default values then click finish Deployment will be under way.



Microsoft.Azure.CosmosDB-20200814100500 | Overview

Deployment is in progress

Deployment name: Microsoft.Azure.CosmosDB-20200814100500
Subscription: Azure subscription 1
Resource group: Azure-Cosmos
Start time: 8/14/2020, 10:35:16 AM
Correlation ID: b6748103-e239-4475-bbc8-53c7960134f1

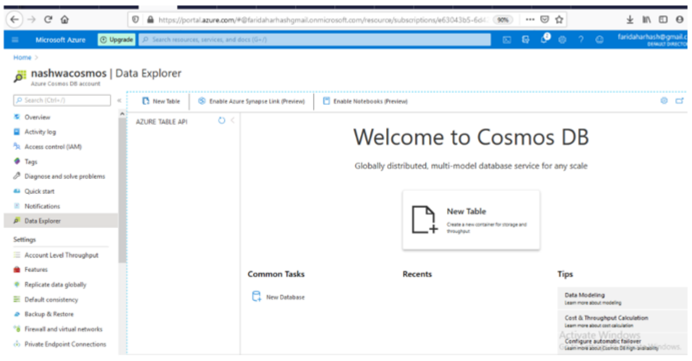
Deployment details (Download)

Resource	Type	Status	Operation details
nashwacosmos	Microsoft.DocumentDb/databa...	OK	Operation details

Figure 4.46 Cosmos DB in Progress

Figure A.45: Cosmos DB in Progress

4. When Deployment is done, you will be redirected to a ”welcome page of Cosmos DB”.



nashwacosmos | Data Explorer

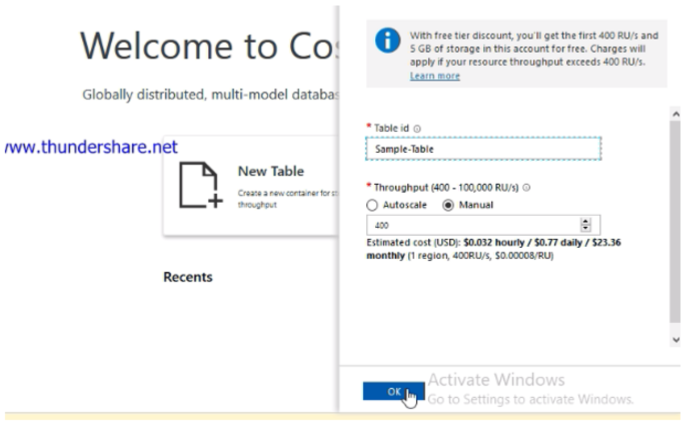
Welcome to Cosmos DB

Globally distributed, multi-model database service for any scale

Figure 4.47 Created Cosmos DB

Figure A.46: Created Cosmos DB

5. Click new table ,then fill the name and throughput you designed for as shown below.



With free tier discount, you'll get the first 400 RU/s and 5 GB of storage in this account for free. Charges will apply if your resource throughput exceeds 400 RU/s. Learn more

* Table id: Sample-Table

* Throughput (400 - 100,000 RU/s): ( ) Autoscale [✓ Manual]

400

Estimated cost (USD): **$0.032 hourly / $0.77 daily / $23.36 monthly** (1 region, 400RU/s, $0.00008/RU)

Welcome to Co
Globally distributed, multi-model databa
www.thundershare.net
New Table
Recents
OK
Activate Windows
Go to Settings to activate Windows.

Figure 4.48 Creating a table

Figure A.47: Creating a table

6. Now you will have this sample table as illustrated below .

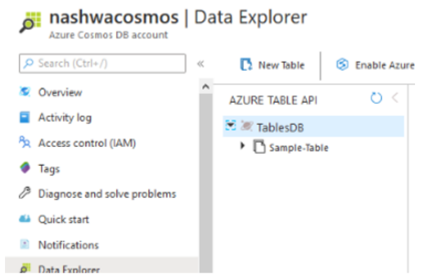


Figure 4.49 Created table

Figure A.48: Created table

7. Click on “Sample Table” to add entities as follow.

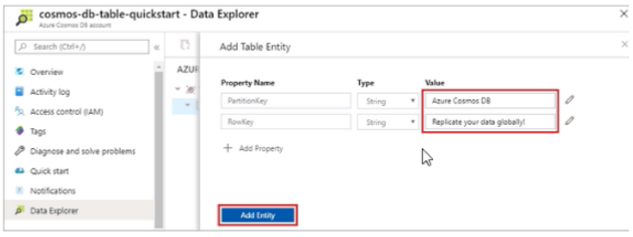
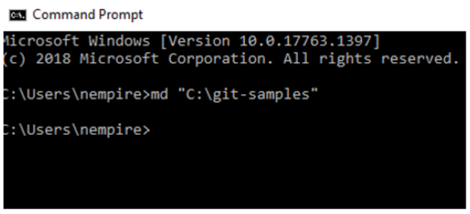


Figure 4.50 Add Entity

Figure A.49: Creating a table

8. Open the command prompt and type md $\backslash c : git{-}samples''tocreateafoldernamed\backslash gitsa$

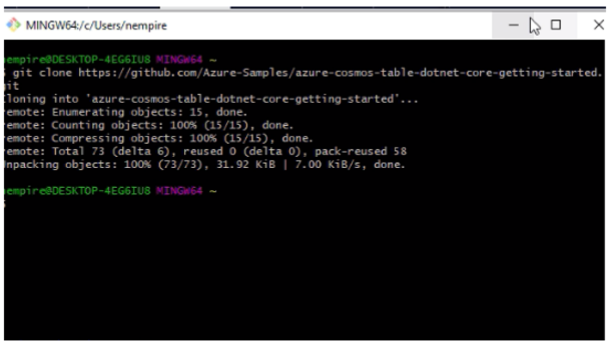


```
icrosoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

:\Users\nempire>md "C:\git-samples"

:\Users\nempire>
```

Figure 4.51 Created folder git samples

Figure A.50: Creating a table

9. Run a git bash, and navigate to “git samples” then type “git clone https://github.com/Azure-Samples/azure-cosmos-table-dotnet-core-getting-started.git” to install sample projects
Note: you have to install get bash first .



```
empire@DESKTOP-4EG6IU8 MINGW64 ~
 git clone https://github.com/Azure-Samples/azure-cosmos-table-dotnet-core-getting-started.
it
loning into 'azure-cosmos-table-dotnet-core-getting-started'...
emote: Enumerating objects: 15, done.
emote: Counting objects: 100% (15/15), done.
emote: Compressing objects: 100% (15/15), done.
emote: Total 73 (delta 6), reused 0 (delta 0), pack-reused 58
npacking objects: 100% (73/73), 31.92 KiB | 7.00 KiB/s, done.

empire@DESKTOP-4EG6IU8 MINGW64 ~
```

Figure 4.52 cloning Sample projects get from GIT repository

Figure A.51: cloning Sample projects get from GIT repository

10. In Visual Studio,click the File menu, select Open, then choose Project/Solution.

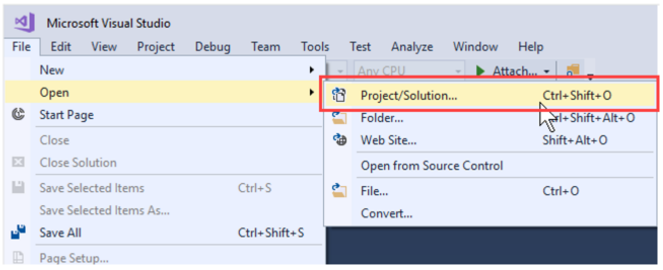


Figure 4.53 Open Project Solution

Figure A.52: Open Project Solution

11. Go to the folder where you copied the sample application and open the TableStorage.sln file

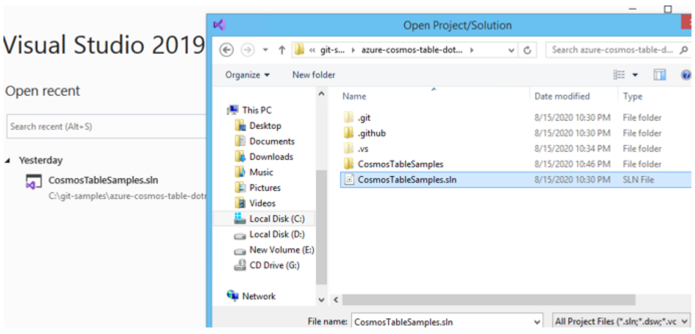
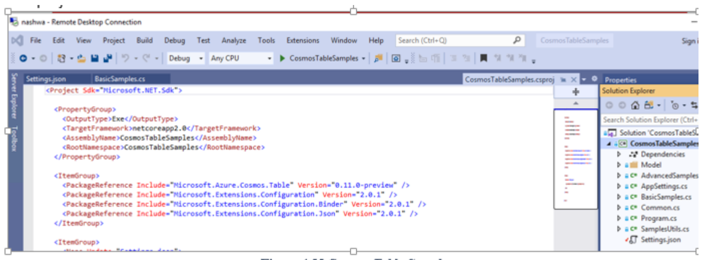

Visual Studio 2019
Open recent
Search recent (Alt+S)
Yesterday
CosmosTableSamples.sln
C:\git-samples\azure-cosmos-table-dot
Open Project/Solution
« git-s... › azure-cosmos-table-dot... ›
Search azure-cosmos-table-d...
Organize
New folder
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
New Volume (E:)
CD Drive (G:)
Network
Name
Date modified
Type
.git 8/15/2020 10:30 PM File folder
.github 8/15/2020 10:30 PM File folder
.vs 8/15/2020 10:34 PM File folder
CosmosTableSamples 8/15/2020 10:46 PM File folder
CosmosTableSamples.sln 8/15/2020 10:30 PM SLN File
File name: CosmosTableSamples.sln
All Project Files (*.sln;*.dsw;*.vc

Figure 4.54 Open Cosmos Samples projects

Figure A.53: Open Cosmos Samples projects

12. The project will be launched.

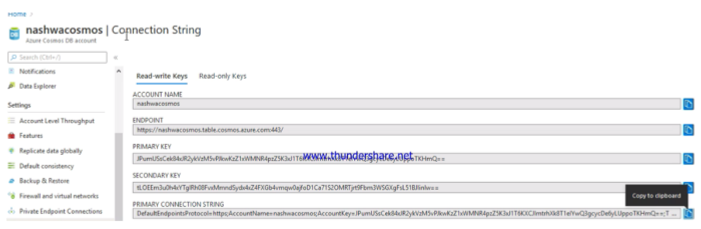


```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp2.0</TargetFramework>
    <AssemblyName>CosmosTableSamples</AssemblyName>
    <RootNamespace>CosmosTableSamples</RootNamespace>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.Azure.Cosmos.Table" Version="0.11.0-preview" />
    <PackageReference Include="Microsoft.Extensions.Configuration" Version="2.0.1" />
    <PackageReference Include="Microsoft.Extensions.Configuration.Binder" Version="2.0.1" />
    <PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="2.0.1" />
  </ItemGroup>

  <ItemGroup>
```

Figure 4.55 Cosmos Table Sample

Figure A.54: Cosmos Table Sample

13. Go to Azure portal, click Connection String. copy the PRIMARY CONNECTION STRING.



nashwacosmos | Connection String

Read-write Keys | Read-only Keys

ACCOUNT NAME: nashwacosmos

ENDPOINT: https://nashwacosmos.table.cosmos.azure.com:443/

PRIMARY KEY: JPumUSsCek84xJR2ykVzM5vPJkwKzZ1xWMNR4pzZ5K3xJ1T6 www.thundershare.net TKHmQ==

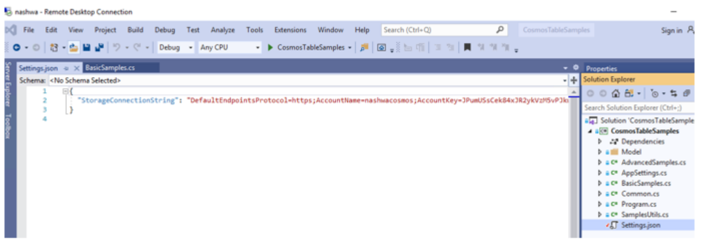
SECONDARY KEY: tLOEEm3u0h4xYTglRh08FvxMmndSydx4xZ4FXGb4vmqw0ajfoD1Ca71S2OMRTjrt9Fbm3WSGXgFsL51BJlinlw==

PRIMARY CONNECTION STRING: DefaultEndpointsProtocol=https;AccountName=nashwacosmos;AccountKey=JPumUSsCek84xJR2ykVzM5vPJkwKzZ1xWMNR4pzZ5K3xJ1T6KXCJlmtrhXk8T1elYwQ3gcycDe6yLUppoTKHmQ==;T ...

Figure 4.56 Connection String

Figure A.55: Connection String

14. Then open the file “settings. json “ under cosmos Table samples solution to paste the primary connection string .



```
{
  "StorageConnectionString": "DefaultEndpointsProtocol=https;AccountName=nashwacosmos;AccountKey=JPumUSsCek84xJR2ykVzM5vPJk
}
```

Figure 4.57 Json Connection String

Figure A.56: Json Connection String

15. Right click on the “CosmosTablesamples” project in solution explorer and select manage nutmeg.

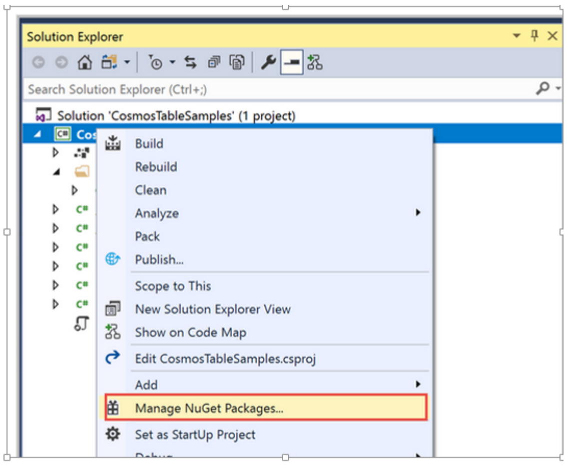
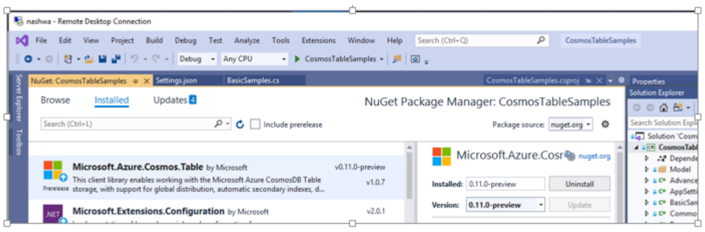


Figure 4.58 NutGEt Packages

Figure A.57: Json Connection String

16. In the NuGet Browse box, type Microsoft.Azure.Cosmos.Table. it will display the Cosmos DB Table API client library.if you enable Azure while installing the visual Studion ,you will find that "Microsoft Azure Cosmos Table" already installed .



NuGet Package Manager: CosmosTableSamples

Microsoft.Azure.Cosmos.Table by Microsoft v0.11.0-preview v1.0.7
This client library enables working with the Microsoft Azure CosmosDB Table storage, with support for global distribution, automatic secondary indexes, d...

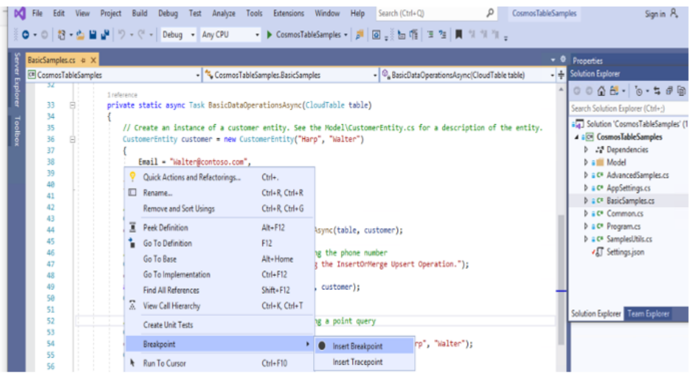
Microsoft.Extensions.Configuration by Microsoft v2.0.1

Microsoft.Azure.Cosr nuget.org
Installed: 0.11.0-preview
Version: 0.11.0-preview

Figure 4.59 Microsoft Azure Cosmos Table Library

Figure A.58: Microsoft Azure Cosmos Table Library

17. Open "Basic Samples.CS" file under COSMOS Table Samples Solution and insert break points on line 52 and 55.just to be able to read data while program is running

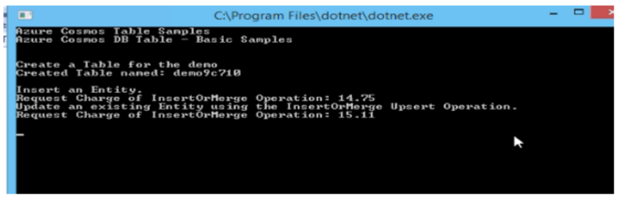


```
        1 reference
33      private static async Task BasicDataOperationsAsync(CloudTable table)
34      {
35          // Create an instance of a customer entity. See the Model\CustomerEntity.cs for a description of the entity.
36          CustomerEntity customer = new CustomerEntity("Harp", "Walter")
37          {
38              Email = "Walter@contoso.com",
44                                                      Async(table, customer);
46                                                      g the phone number
47                                                      g the InsertOrMerge Upsert Operation.");
49                                                      , customer);
52                                                      g a point query
54                                                      rp", "Walter");
```

Figure 4.60 Insert Break point

Figure A.59: Insert Break point

18. Press F5 to execute the application ,the console will show the name of the new table which is DEMOA13B1 in aure CosmosDB



C:\Program Files\dotnet\dotnet.exe

```
Azure Cosmos Table Samples
Azure Cosmos DB Table - Basic Samples

Create a Table for the demo
Created Table named: demo9c710

Insert an Entity.
Request Charge of InsertOrMerge Operation: 14.75
Update an existing Entity using the InsertOrMerge Upsert Operation.
Request Charge of InsertOrMerge Operation: 15.11

_
```

Figure 4.61 Running Application

Figure A.60: Running Application

19. Also go to Azure portal to show the created table by the running app.

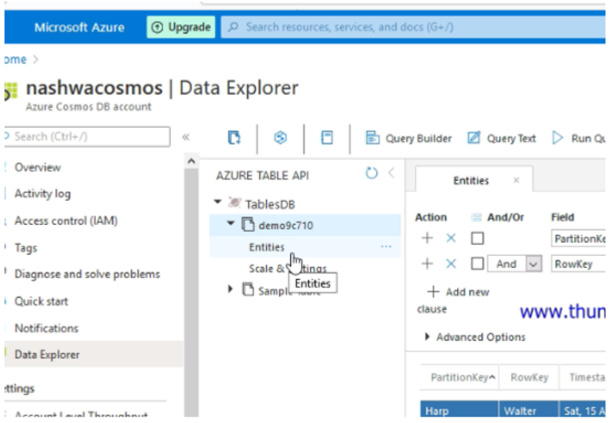


Figure 4.62 Table created by Application

Figure A.61: Table created by Application

20. Also navigate to show the throughput.

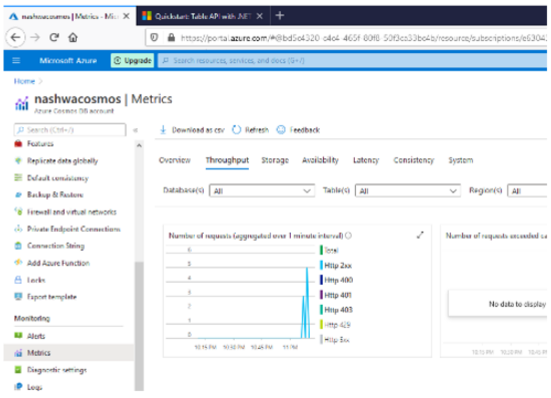
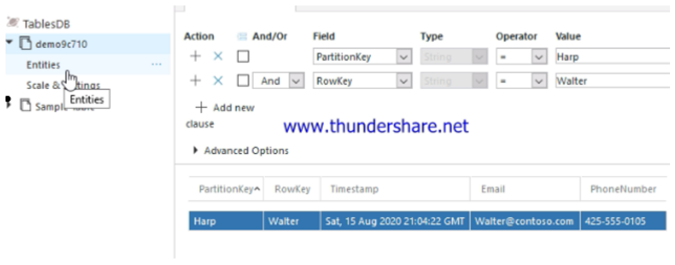


Figure 4.63 Number of Http request

Figure A.62: Number of Http request

21. Press F5 to run the app to the next breakpoint. When you hit the breakpoint, switch back to the Azure portal, click Entities again to open the Entities tab, and note that the phone number has been updated to 425-555-0105.



Action	And/Or	Field	Type	Operator	Value
+ × ☐		PartitionKey	String	=	Harp
+ × ☐	And	RowKey	String	=	Walter

+ Add new clause

www.thundershare.net

Advanced Options

PartitionKey	RowKey	Timestamp	Email	PhoneNumber
Harp	Walter	Sat, 15 Aug 2020 21:04:22 GMT	Walter@contoso.com	425-555-0105

Figure 4.64 phone number value

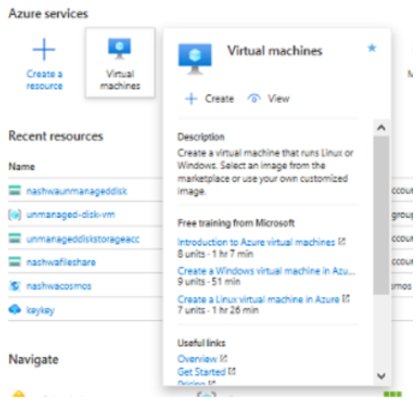
Figure A.63: phone number value

22. Press F5 to run the app. The app adds entities for use in an advanced sample app that the Table API currently does not support. The app then deletes the table created by the sample app. In the console window, press Enter to end the execution of the app.

A.8 Creating a Virtual Machine with Unmanaged Disk

In order to create a VM with unmanaged Disk, you have to create a storage account first

1. Navigate to Azure portal and select virtual machine ,create a virtual machine



Description
Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.

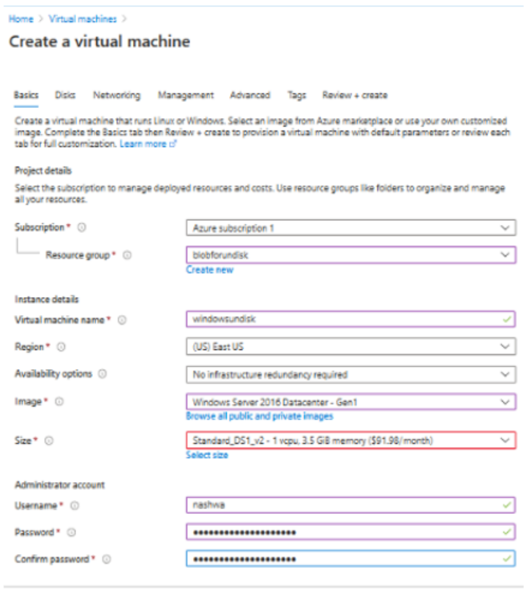
Free training from Microsoft
Introduction to Azure virtual machines
8 units · 1 hr 7 min
Create a Windows virtual machine in Azu...
9 units · 51 min
Create a Linux virtual machine in Azure
7 units · 1 hr 26 min

Useful links
Overview
Get Started

Figure 4.65 Virtual Machine Menu 1

Figure A.64: Virtual Machine Menu 1

2. The following page will be displayed asking about resource group ,machine name ,region,image, size, username and password



Home > Virtual machines >

Create a virtual machine

Basics | Disks | Networking | Management | Advanced | Tags | Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Learn more

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Azure subscription 1
Resource group *	blobforundisk (Create new)

Instance details

Virtual machine name *	windowsundisk
Region *	(US) East US
Availability options	No infrastructure redundancy required
Image *	Windows Server 2016 Datacenter - Gen1 (Browse all public and private images)
Size *	Standard_DS1_v2 - 1 vcpu, 3.5 GiB memory ($91.98/month) (Select size)

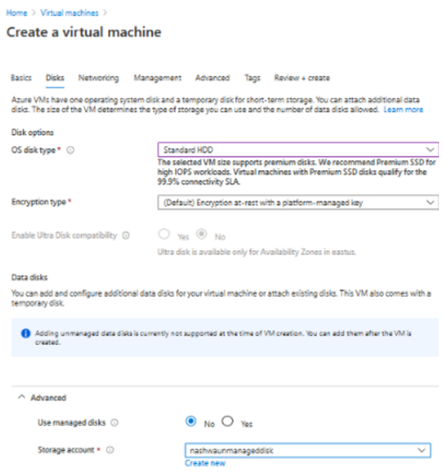
Administrator account

Username *	nashwa
Password *	••••••••••••••••••••
Confirm password *	••••••••••••••••••••

Figure 4.66 Virtual Machine Menu

Figure A.65: Virtual Machine Menu

3. In Disk Section select standard HDD Disk and press radio button no for use managed disk and select Storage account "nashwaunmanagedDisk" which was created in advance.



Home > Virtual machines >

Create a virtual machine

Basics | Disks | Networking | Management | Advanced | Tags | Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. Learn more

Disk options

OS disk type *	Standard HDD
	The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.
Encryption type *	(Default) Encryption at-rest with a platform-managed key
Enable Ultra Disk compatibility	Yes [✓ No]
	Ultra disk is available only for Availability Zones in eastus.

Data disks

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

Adding unmanaged data disks is currently not supported at the time of VM creation. You can add them after the VM is created.

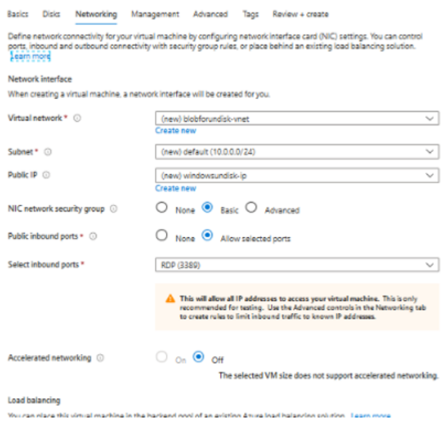
Advanced

Use managed disks	[✓ No] Yes
Storage account *	nashwaunmanageddisk (Create new)

Figure 4.67 Creating VM with unmanaged disk

Figure A.66: Virtual Machine Menu

4. Accept the default of the rest section "network", "management"


Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network *	(new) blobforundisk-vnet Create new
Subnet *	(new) default (10.0.0.0/24)
Public IP	(new) windowsundisk-ip Create new
NIC network security group	None [✓ Basic] Advanced
Public inbound ports *	None [✓ Allow selected ports]
Select inbound ports *	RDP (3389)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

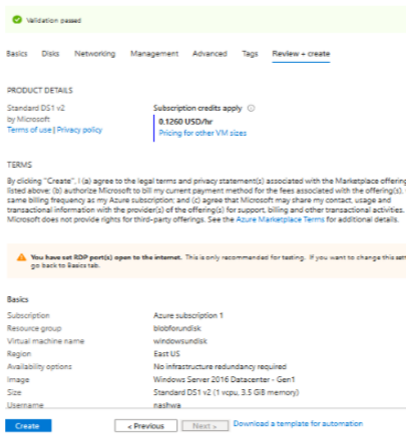
Accelerated networking	On [✓ Off] The selected VM size does not support accelerated networking.

Load balancing

Figure 4.68 Creating VM Network Setting

Figure A.67: Creating VM Network Setting

5. Then select create


Validation passed

Basics Disks Networking Management Advanced Tags Review + create

PRODUCT DETAILS

Standard DS1 v2
by Microsoft
Terms of use | Privacy policy

Subscription credits apply
0.1260 USD/hr
Pricing for other VM sizes

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the Azure Marketplace Terms for additional details.

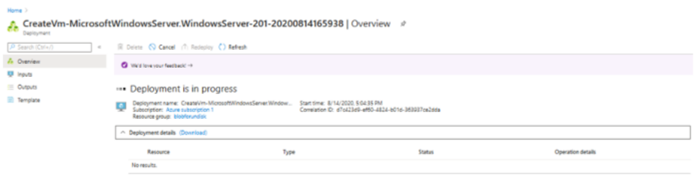
You have set RDP port(s) open to the internet. This is only recommended for testing. If you want to change this set go back to Basics tab.

Basics

Subscription	Azure subscription 1
Resource group	blobforundisk
Virtual machine name	windowsundisk
Region	East US
Availability options	No infrastructure redundancy required
Image	Windows Server 2016 Datacenter - Gen1
Size	Standard DS1 v2 (1 vcpu, 3.5 GiB memory)
Username	nashwa

Figure 4.69 Creating VM Validation

Figure A.68: Creating VM Validation



CreateVm-MicrosoftWindowsServer.WindowsServer-201-20200814165938 | Overview
Deployment

Deployment is in progress

Deployment name: CreateVm-MicrosoftWindowsServer.Window...
Subscription: Azure subscription 1
Resource group: blobforundisk
Start time: 8/14/2020, 5:04:35 PM
Correlation ID: d7c423d9-ef60-4824-b01d-363937ca2dda

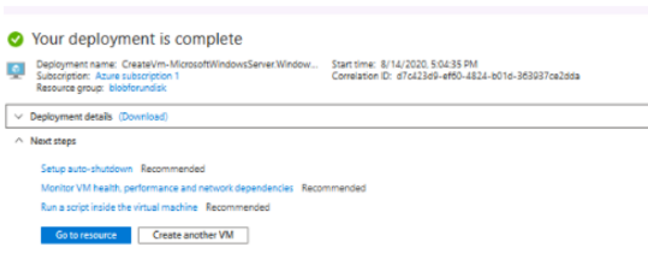
Deployment details (Download)

Resource	Type	Status	Operation details
No results.			

Figure 4.70 VM Deployment in progress

Figure A.69: VM Deployment in progress

6. When you see your deployment is complete as below :


Your deployment is complete

Deployment name: CreateVm-MicrosoftWindowsServer.Window...
Subscription: Azure subscription 1
Resource group: blobforundisk

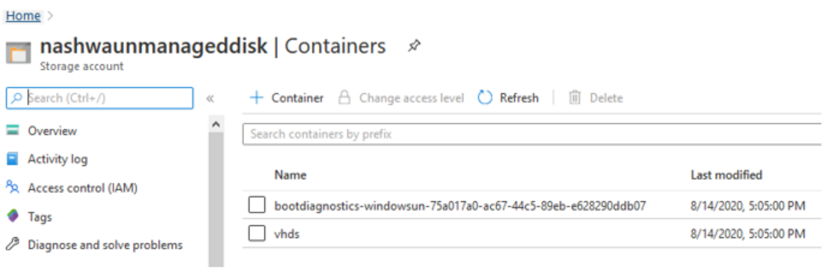
Start time: 8/14/2020, 5:04:35 PM
Correlation ID: d7c423d9-ef60-4824-b01d-363937ca2dda

Deployment details (Download)
Next steps
Setup auto-shutdown Recommended
Monitor VM health, performance and network dependencies Recommended
Run a script inside the virtual machine Recommended
Go to resource
Create another VM

Figure 4.71 VM Deployment finished

Figure A.70: VM Deployment finished

7. Navigate to storage account called "Nashwa unmanaged Disk".you will notice that blob containers has the VHD container

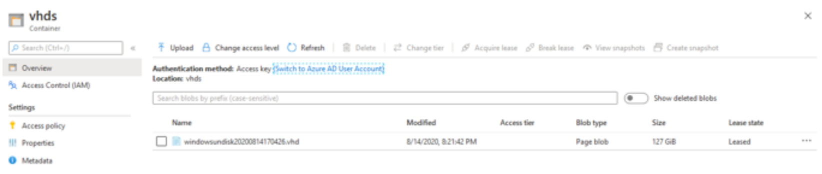


Name	Last modified
bootdiagnostics-windowsun-75a017a0-ac67-44c5-89eb-e628290ddb07	8/14/2020, 5:05:00 PM
vhds	8/14/2020, 5:05:00 PM

Figure 4.72 VM Hard disk in Blob

Figure A.71: VM Hard disk in Blob

8. Now you can see the virtual hard disk in the vhd blob.



Authentication method: Access key (Switch to Azure AD User Account)
Location: vhds

Name	Modified	Access tier	Blob type	Size	Lease state
windowsundisk20200814170426.vhd	8/14/2020, 8:21:42 PM		Page blob	127 GiB	Leased

Figure 4.73 VM VHD in Blob

Figure A.72: VM VHD in Blob

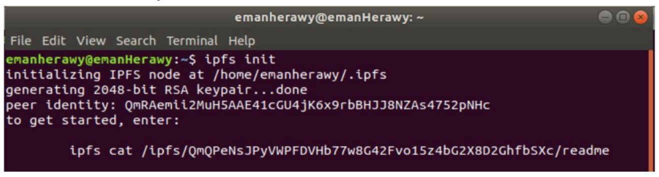
A Appendix C:IPFS

This appendix contains details about how to actually connect to IPFS as an external node and as an embedded node.

1. As external node **To start, you'll need to install it on your system:**
    - Install IPFS https://dist.IPFS.io/go-IPFS

65

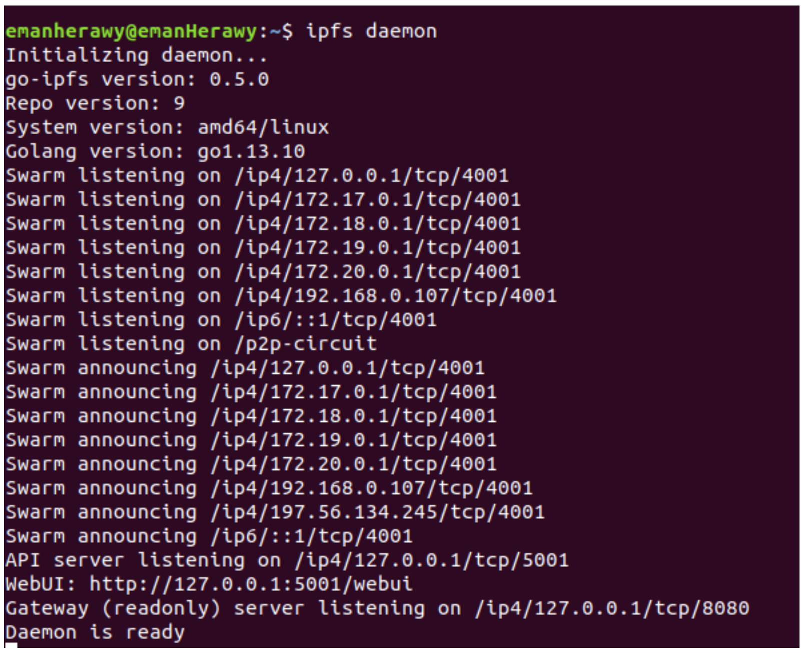
- Create node identity



```
emanherawy@emanHerawy:~$ ipfs init
initializing IPFS node at /home/emanherawy/.ipfs
generating 2048-bit RSA keypair...done
peer identity: QmRAemii2MuH5AAE41cGU4jK6x9rbBHJJ8NZAs4752pNHc
to get started, enter:

        ipfs cat /ipfs/QmQPeNsJPyVWPFDVHb77w8G42Fvo15z4bG2X8D2GhfbSXc/readme
```

Figure A.1: Create node identity

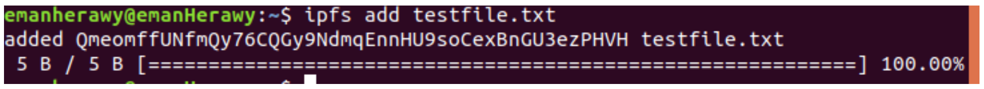
- Take your node online.



```
emanherawy@emanHerawy:~$ ipfs daemon
Initializing daemon...
go-ipfs version: 0.5.0
Repo version: 9
System version: amd64/linux
Golang version: go1.13.10
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/172.17.0.1/tcp/4001
Swarm listening on /ip4/172.18.0.1/tcp/4001
Swarm listening on /ip4/172.19.0.1/tcp/4001
Swarm listening on /ip4/172.20.0.1/tcp/4001
Swarm listening on /ip4/192.168.0.107/tcp/4001
Swarm listening on /ip6/::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/172.17.0.1/tcp/4001
Swarm announcing /ip4/172.18.0.1/tcp/4001
Swarm announcing /ip4/172.19.0.1/tcp/4001
Swarm announcing /ip4/172.20.0.1/tcp/4001
Swarm announcing /ip4/192.168.0.107/tcp/4001
Swarm announcing /ip4/197.56.134.245/tcp/4001
Swarm announcing /ip6/::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
```

Figure A.2: Take your node online.

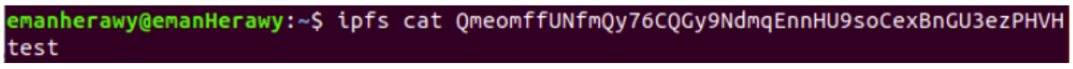
- By default, your gateway is not exposed to the world. It only works locally. To add file , IPFS add ¡file¿



```
emanherawy@emanHerawy:~$ ipfs add testfile.txt
added QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHVH testfile.txt
 5 B / 5 B [============================================================] 100.00%
```

Figure A.3: Add file from cli

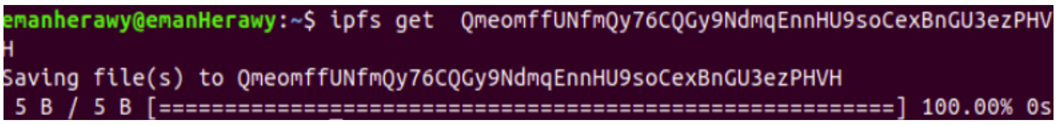
- To read file , IPFS cat ¡CID¿



```
emanherawy@emanHerawy:~$ ipfs cat QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHVH
test
```

Figure A.4: Add cat file in cli

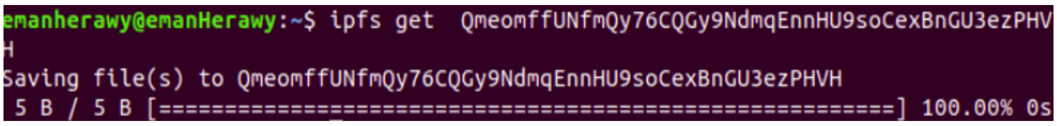
- To download the file : IPFS get ¡CID¿



```
emanherawy@emanHerawy:~$ ipfs get  QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHV
H
Saving file(s) to QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHVH
 5 B / 5 B [==========================================================] 100.00% 0s
```

Figure A.5: IPFS get

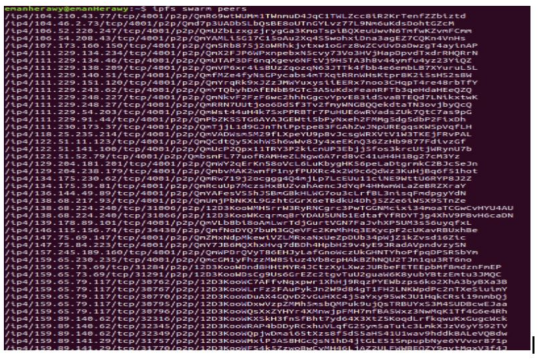
- To list all current peers : IPFS swarm peers.



```
emanherawy@emanHerawy:~$ ipfs get  QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHV
H
Saving file(s) to QmeomffUNfmQy76CQGy9NdmqEnnHU9soCexBnGU3ezPHVH
 5 B / 5 B [==========================================================] 100.00% 0s
```

Figure A.6: IPFS get

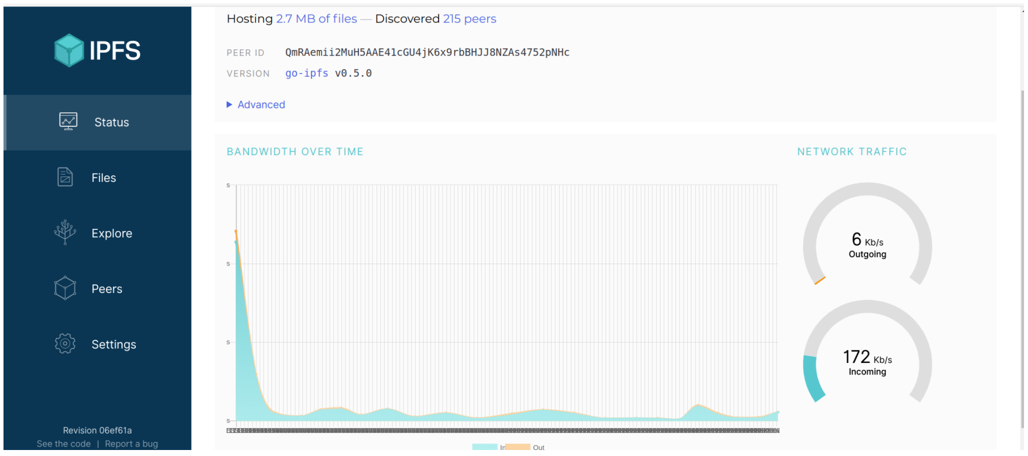
2. To list all current peers : IPFS swarm peers.



```
emanherawy@emanHerawy:~$ ipfs swarm peers
/ip4/104.210.43.77/tcp/4001/p2p/QmR69wtWUMm1TWnmuD4JqC1TWLZcc8iR2KrTenfZZbiztd
/ip4/104.46.2.73/tcp/4001/p2p/Qmd7p3UADbSLbQsBE8oUTnGYLvz77L9Nm6uKdsDohtGZcM
/ip4/106.52.220.247/tcp/4001/p2p/QmUZbLzxgzjrygGa3KmoTspiBQXeuUwvN6TmfwKZvmFCmm
/ip4/106.54.208.43/tcp/4001/p2p/QmYAMLi5G17C15oAu2Xq4S5wohxiDna3agEZ7CQKn4VnHs
/ip4/107.173.160.150/tcp/4001/p2p/QmSRb875j2oWRhkjvtxw1oGrz8wZCvUvDaDwzgT4ayinAP
/ip4/111.229.134.234/tcp/4001/p2p/QmX2FJP6WPxnpebxN5cvy73Vo3HVjHapDpvdTxdrRHQRrN
/ip4/111.229.134.46/tcp/4001/p2p/QmUTAP3DF6nqXgev6NFtVj9H5TA3h8v44ymfu4yz23YiQZ
/ip4/111.229.138.209/tcp/4001/p2p/QmVP6xr4isBUzZqozqN63JTTk4fbb4e6embLB7XYuruL5L
/ip4/111.229.140.51/tcp/4001/p2p/QmfMZe4fyNsGPycabs4mTXqtRRnWHsKtpr8K2i5sHS2s8W
/ip4/111.229.151.120/tcp/4001/p2p/QmYrqRk9xJZzJMWYuxysilEERx7noo3CHqpT4re48rbTfY
/ip4/111.229.243.62/tcp/4001/p2p/QmYTQbyhDAfENbB9GTc3A5uKdxFeanRFTb3qeHdaHEeQZQ
/ip4/111.229.248.227/tcp/4001/p2p/QmNkvF2FzF6wc2hhhGgcvYpvE83idSvaBTEQd7LNikxtwK
/ip4/111.229.248.27/tcp/4001/p2p/QmRRN7UUtjoo6Dd5f3Tv2fmyWNGBQQekdtaTN3ovjbyQcQ
/ip4/111.229.54.203/tcp/4001/p2p/QmWst44uH4k7SxPPRBTr7PuHUE6wRVadsZUk7QtC7as9pG
/ip4/111.229.91.44/tcp/4001/p2p/QmPbZKSSTG6AYAJGEWti5bPyNxeh2FMMg5dg5dbP2FixDh
/ip4/111.230.173.37/tcp/4001/p2p/QmTjjL1d9GJnThiPptpe83FGAhZwJNpUREgqsKWSpVqfLH
/ip4/118.25.235.214/tcp/4001/p2p/QmVADWsmSM29fLXpeYU9p8vJcsgWRXVtV1W3TKEjFRvPAL
/ip4/122.51.11.123/tcp/4001/p2p/QmQCdtQy5XxhWSh6wWv8Jy4xeEKnQ36ZzHb9877FdivzGf
/ip4/122.51.141.100/tcp/4001/p2p/QmUcP2Qpx11TRY3P2kicnUP3Ebjj5fos3krcUtjWRynU7b
/ip4/122.51.52.79/tcp/4001/p2p/QmbsmFL77uofRAMHeZLNgw6A7rd8vC41uH4H1Bg27cM3Yz
/ip4/129.204.181.201/tcp/4001/p2p/QmWY2qErKn58oVcL6LuKbygHK56peLaDtgrmkC2BJcSeJn
/ip4/129.204.238.179/tcp/4001/p2p/QmbvMAK2wmfP1nyfPUXRc4x2W9c6QdWz3KuHjBq6f51hot
/ip4/134.175.230.62/tcp/4001/p2p/QmRw719j2ocggg4Q4mjLp7LcEUu11ciNE9WttU6RYP8J2Z
/ip4/134.175.39.81/tcp/4001/p2p/QmRcuUp7MczsHxBUZvahAencJdYqP4HHwmWLaZeBRZXraY
/ip4/136.144.49.89/tcp/4001/p2p/QmYAFesV55hJ5BmGBkHLWG7ou3cLrfBL3nisqFmdpgyYdN
/ip4/138.68.217.93/tcp/4001/p2p/QmUmjPbNKXL9GzhtGGrX6eTBdkU4DhjSZZe6iW5X9STnZe
/ip4/138.68.224.240/tcp/31006/p2p/12D3KooWMH5rrW3RyRNCgrc3PwTGGNMcixi34moaTCGwCvHYU4AU
/ip4/138.68.224.240/tcp/31066/p2p/12D3KooWKcqrmqBrYDAU5UNb1EdtafYfRDYTjg4XhV9PBvH6caDN
/ip4/139.178.89.101/tcp/4001/p2p/QmVLb8bi8oAmLwrTdjGurtVGN7FaJvhXP5UM3s56uyqfxL
/ip4/146.115.156.74/tcp/34430/p2p/QmfNoDYQ7buM3GQeVFc2KmMhHq3EKycpF2cUKavRBUxhBe
/ip4/147.75.69.147/tcp/4001/p2p/QmZMxNdpMkewiVZLMRxaNxUeZpDUb34pWjZ1kZvsd16Zic
/ip4/147.75.84.223/tcp/4001/p2p/QmY7JB6MQXhxHvq7dBDh4HpbH29v4yE9JRadAVpndvzySN
/ip4/157.245.189.160/tcp/4001/p2p/QmWPDrQVyTB6EHJyLafGnoWczUkGHNTYhoPfpqDPSRSbYm
/ip4/159.65.230.235/tcp/4001/p2p/QmcGM1yFhzzMW8Siuz4Vb8cpHAkBZhNQU2TJn1qu3RT6no
/ip4/159.65.73.69/tcp/31284/p2p/12D3KooWDnd8HHtMYR4JCtzXyLXwzJURbeFETEEpbMf8mdznFmEP
/ip4/159.65.73.69/tcp/31291/p2p/12D3KooWDsCg9Us6GrEZc2tgvTuU2guaW6K8yubYBtzEmtu3JMQC
/ip4/159.65.79.117/tcp/30762/p2p/12D3KooWC7AFfvNqxpwr1XhHj9RqzPYEWbzps6ko2XhA3byBXa3B
/ip4/159.65.79.117/tcp/30767/p2p/12D3KooWLrFz2FAuPykJn2W9d84gT1FH2LNKWpdPc2nTXeSiuimY
/ip4/159.65.79.117/tcp/30770/p2p/12D3KooWDuAX4GQvD2vGuHXC4j5aYxy95wKJU1HqkCRsi19nmbQj
/ip4/159.65.79.117/tcp/30795/p2p/12D3KooWDxwVzpZMMh5msbQMPuk9ujQsTRBUYxS3M4SUDBcwEJaa
/ip4/159.65.79.117/tcp/30796/p2p/12D3KooWQsXxZYHYr4XMnwjpFMH7mfBA5Wxz3NwMqK1Tf4G6e4Rh
/ip4/159.89.140.62/tcp/32316/p2p/12D3KooWKXSkH3fnSfBht7yd64X3XtZ5KoqdLrfkqwuKxGugcWck
/ip4/159.89.140.62/tcp/32345/p2p/12D3KooWRAP4bDDyRCxhuVLqfG2Sym5aTuic3LmkXJzV6yY592TV
/ip4/159.89.140.62/tcp/32349/p2p/12D3KooWQpjwDmai65tXzs8f5dS5aHS41U1wav9hddkBALeVQBdw
/ip4/159.89.141.29/tcp/31757/p2p/12D3KooWMxiPJAS8HGcQsN1hD4jtGLE51SmpupbNye6YVvor871p
/ip4/159.89.141.29/tcp/31770/p2p/12D3KooWFS4k5ZzwoBwCvMH46LjAZZULFWWBEQZY9qvtMqxV3f4J
```

Figure A.7: IPFS get

3. Manage your node in Web console : open http://127.0.0.1:5001/webui You can



Hosting 2.7 MB of files — Discovered 215 peers

PEER ID QmRAemii2MuH5AAE41cGU4jK6x9rbBHJJ8NZAs4752pNHc

VERSION go-ipfs v0.5.0

Figure A.8: Dashboard

get some states about your current bandwidth, files ,and connected peers , their locations and latency .

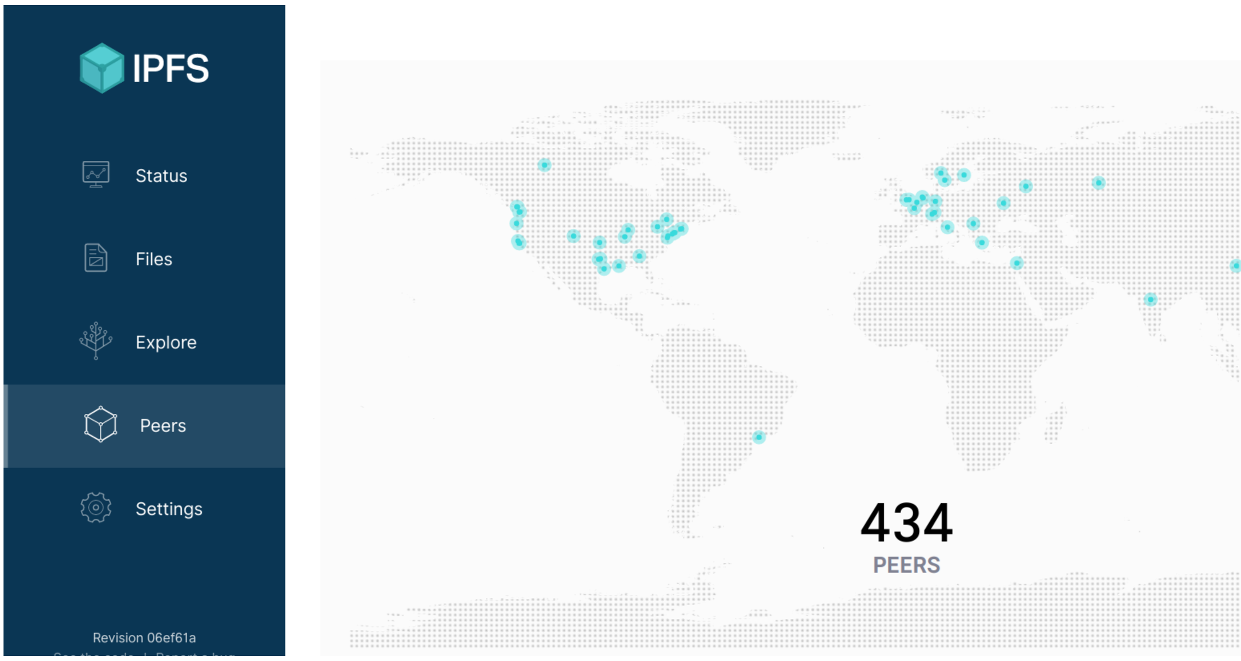
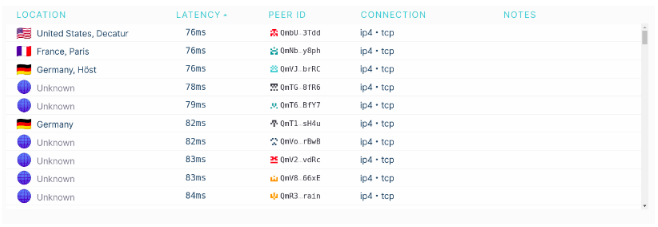


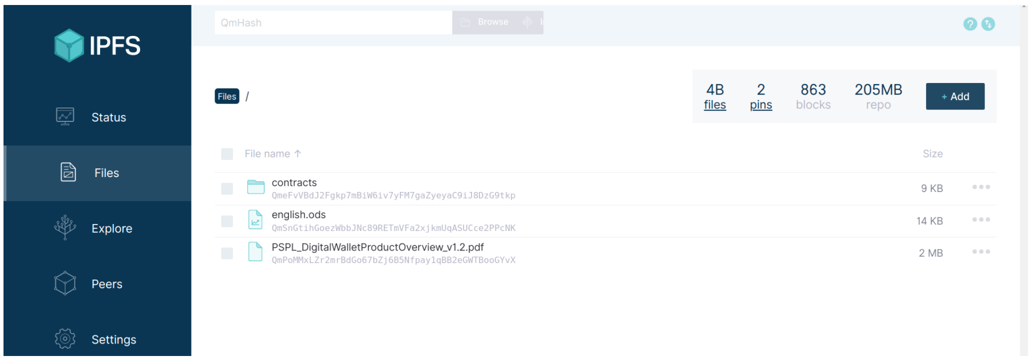
Figure A.9: IPFS Peers



LOCATION	LATENCY ▴	PEER ID	CONNECTION	NOTES
United States, Decatur	76ms	QmbU…3Tdd	ip4 • tcp	
France, Paris	76ms	QmNb…y8ph	ip4 • tcp	
Germany, Höst	76ms	QmVJ…brRC	ip4 • tcp	
Unknown	78ms	QmTG…8fR6	ip4 • tcp	
Unknown	79ms	QmT6…BfY7	ip4 • tcp	
Germany	82ms	QmT1…sH4u	ip4 • tcp	
Unknown	82ms	QmVo…rBwB	ip4 • tcp	
Unknown	83ms	QmV2…vdRc	ip4 • tcp	
Unknown	83ms	QmV8…66xE	ip4 • tcp	
Unknown	84ms	QmR3…rain	ip4 • tcp	

Figure A.10: peer by country

4. After 5 minutes

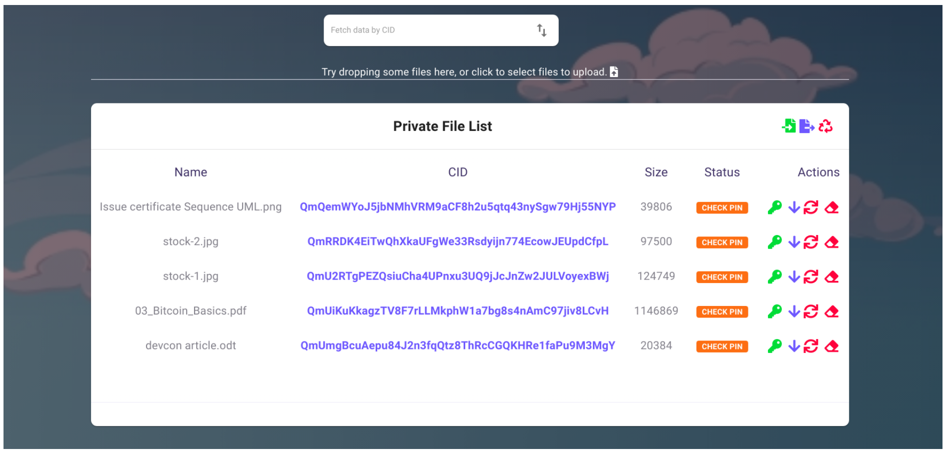


4B files	2 pins	863 blocks	205MB repo

File name ↑	Size
contracts QmeFvVBdJ2Fgkp7mBiW6iv7yFM7gaZyeyaC9iJ8DzG9tkp	9 KB
english.ods QmSnGtihGoezWbbJNc89RETmVFa2xjkmUqASUCce2PPcNK	14 KB
PSPL_DigitalWalletProductOverview_v1.2.pdf QmPoMMxLZr2mrBdGo67bZj6B5Nfpay1qBB2eGWTBooGYvX	2 MB

Figure A.11: file in web console

AS Embedded node For testing purposes, we have developed a basic react app to connect as an embedded node. We have developed an AES based encryption/decryption function in order to add the missed built-in privacy function. https://github.com/EmanHerawy/web

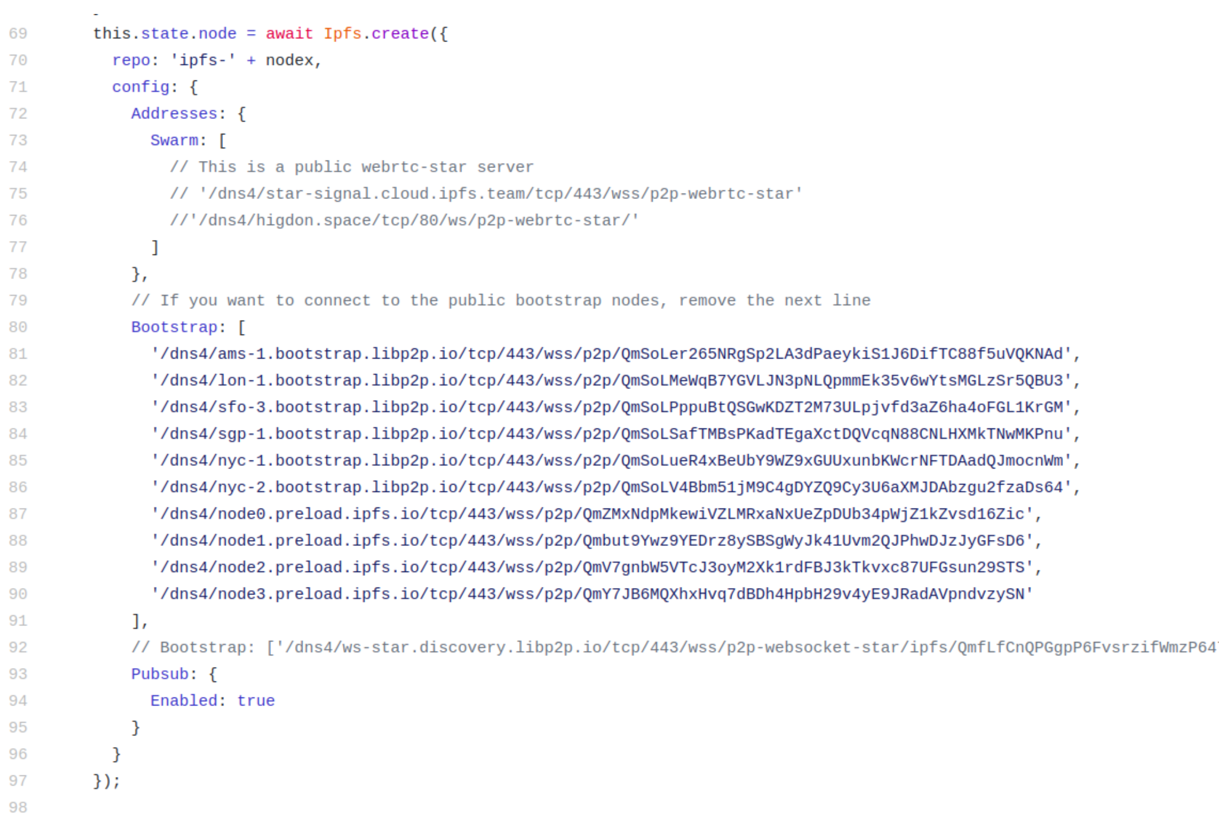

Fetch data by CID
Try dropping some files here, or click to select files to upload.

Private File List

Name	CID	Size	Status	Actions
Issue certificate Sequence UML.png	QmQemWYoJ5jbNMhVRM9aCF8h2u5qtq43nySgw79Hj55NYP	39806	CHECK PIN	
stock-2.jpg	QmRRDK4EiTwQhXkaUFgWe33Rsdyijn774EcowJEUpdCfpL	97500	CHECK PIN	
stock-1.jpg	QmU2RTgPEZQsiuCha4UPnxu3UQ9jJcJnZw2JULVoyexBWj	124749	CHECK PIN	
03_Bitcoin_Basics.pdf	QmUiKuKkagzTV8F7rLLMkphW1a7bg8s4nAmC97jiv8LCvH	1146869	CHECK PIN	
devcon article.odt	QmUmgBcuAepu84J2n3fqQtz8ThRcCGQKHRe1faPu9M3MgY	20384	CHECK PIN	

Figure A.12: running app

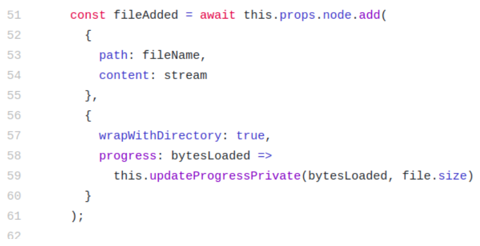
- Add js-IPFS backpage Using Npm , npm install IPFS Browser CDN $< scriptsrc = ”https : //cdn.jsdelivr.net/npm/IPFS/dist/index.min.js” >< /script >$
- To start node and connect to IPFS through public nodes .



```
this.state.node = await Ipfs.create({
  repo: 'ipfs-' + nodex,
  config: {
    Addresses: {
      Swarm: [
        // This is a public webrtc-star server
        // '/dns4/star-signal.cloud.ipfs.team/tcp/443/wss/p2p-webrtc-star'
        //'/dns4/higdon.space/tcp/80/ws/p2p-webrtc-star/'
      ]
    },
    // If you want to connect to the public bootstrap nodes, remove the next line
    Bootstrap: [
      '/dns4/ams-1.bootstrap.libp2p.io/tcp/443/wss/p2p/QmSoLer265NRgSp2LA3dPaeykiS1J6DifTC88f5uVQKNAd',
      '/dns4/lon-1.bootstrap.libp2p.io/tcp/443/wss/p2p/QmSoLMeWqB7YGVLJN3pNLQpmmEk35v6wYtsMGLzSr5QBU3',
      '/dns4/sfo-3.bootstrap.libp2p.io/tcp/443/wss/p2p/QmSoLPppuBtQSGwKDZT2M73ULpjvfd3aZ6ha4oFGL1KrGM',
      '/dns4/sgp-1.bootstrap.libp2p.io/tcp/443/wss/p2p/QmSoLSafTMBsPKadTEgaXctDQVcqN88CNLHXMkTNwMKPnu',
      '/dns4/nyc-1.bootstrap.libp2p.io/tcp/443/wss/p2p/QmSoLueR4xBeUbY9WZ9xGUUxunbKWcrNFTDAadQJmocnWm',
      '/dns4/nyc-2.bootstrap.libp2p.io/tcp/443/wss/p2p/QmSoLV4Bbm51jM9C4gDYZQ9Cy3U6aXMJDAbzgu2fzaDs64',
      '/dns4/node0.preload.ipfs.io/tcp/443/wss/p2p/QmZMxNdpMkewiVZLMRxaNxUeZpDUb34pWjZ1kZvsd16Zic',
      '/dns4/node1.preload.ipfs.io/tcp/443/wss/p2p/Qmbut9Ywz9YEDrz8ySBSgWyJk41Uvm2QJPhwDJzJyGFsD6',
      '/dns4/node2.preload.ipfs.io/tcp/443/wss/p2p/QmV7gnbW5VTcJ3oyM2Xk1rdFBJ3kTkvxc87UFGsun29STS',
      '/dns4/node3.preload.ipfs.io/tcp/443/wss/p2p/QmY7JB6MQXhxHvq7dBDh4HpbH29v4yE9JRadAVpndvzySN'
    ],
    // Bootstrap: ['/dns4/ws-star.discovery.libp2p.io/tcp/443/wss/p2p-websocket-star/ipfs/QmfLfCnQPGgpP6FvsrzifWmzP64
    Pubsub: {
      Enabled: true
    }
  }
});
```

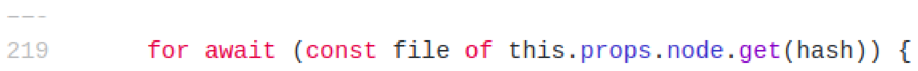
Figure A.13: create ipfs node in js

- To add file to IPFS
  public nodes .
- To fetch (get data)



```
const fileAdded = await this.props.node.add(
  {
    path: fileName,
    content: stream
  },
  {
    wrapWithDirectory: true,
    progress: bytesLoaded =>
      this.updateProgressPrivate(bytesLoaded, file.size)
  }
);
```

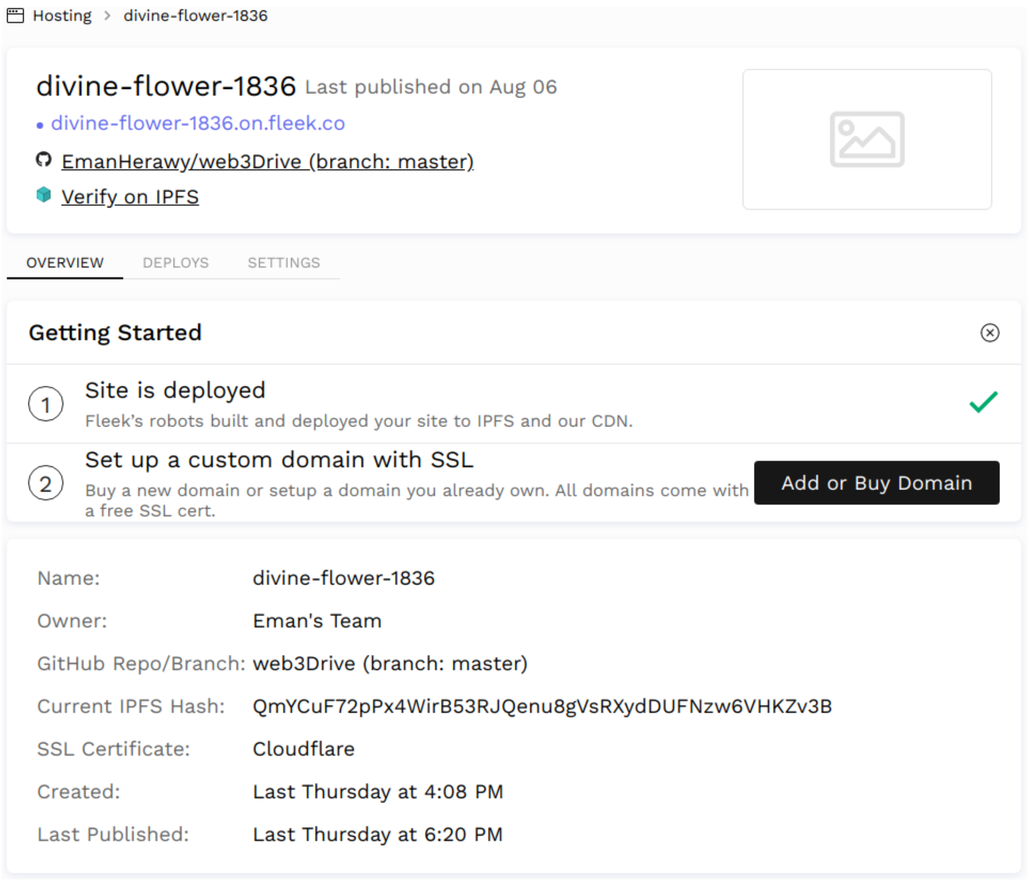
Figure A.14: ipfs add in js



```
for await (const file of this.props.node.get(hash)) {
```

Figure A.15: ipfs get in js

In order to explore the free hosting features that IPFS offer, we have hosted the app using Fleek and got a free domain linked to CID based on IPFS IPNS.



Hosting > divine-flower-1836

divine-flower-1836 Last published on Aug 06

• divine-flower-1836.on.fleek.co

EmanHerawy/web3Drive (branch: master)

Verify on IPFS

Getting Started

1 Site is deployed
Fleek's robots built and deployed your site to IPFS and our CDN.

2 Set up a custom domain with SSL
Buy a new domain or setup a domain you already own. All domains come with a free SSL cert.

Name:	divine-flower-1836
Owner:	Eman's Team
GitHub Repo/Branch:	web3Drive (branch: master)
Current IPFS Hash:	QmYCuF72pPx4WirB53RJQenu8gVsRXydDUFNzw6VHKZv3B
SSL Certificate:	Cloudflare
Created:	Last Thursday at 4:08 PM
Last Published:	Last Thursday at 6:20 PM

Figure A.16: ihosting in fleek

References

[1] "Grant limited access to data with shared access signatures (sas) - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview, (Accessed on 08/19/2020).

[2] "Microsoft azure data encryption-at-rest — microsoft docs," https://docs.microsoft.com/en-us/azure/security/fundamentals/encryption-atrest, (Accessed on 08/19/2020).

[3] D. Technologies, *Information Storage and Management Version 4.0*. Dell Technologies, 2019.

[4] "Not found," https://www.mvpindex.com/go/white-paper-marketers-guide-last-minute-holiday-data-collection, (Accessed on 08/22/2020).

[5] "Executive summary: Data growth, business opportunities, and the it imperatives — the digital universe of opportunities: Rich data and the increasing value of the internet of things," https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm, (Accessed on 08/22/2020).

[6] "Guidelines for building a private cloud infrastructure," http://nexgsd.org/wp-content/uploads/2012/05/Guidelines-to-BuildingPrivateCloud-Infrastructure-Technical-Report.pdf, (Accessed on 08/14/2020).

[7] "What is a data center? - palo alto networks," https://www.paloaltonetworks.com/cyberpedia/what-is-a-data-center, (Accessed on 08/14/2020).

[8] "Nas vs. san vs. das: Which is right for you? — seagate blog," https://blog.seagate.com/business/nas-vs-san-vs-das-which-is-right-for-you/, (Accessed on 08/15/2020).

[9] "What is software-defined infrastructure? sdi explained – bmc blogs," https://www.bmc.com/blogs/software-defined-infrastructure/, (Accessed on 08/15/2020).

[10] "What is data center orchestration? definition & faqs — avi networks," https://avinetworks.com/glossary/data-center-orchestration/#:~:text=Data%20Center%20Orchestration%20Definition,FAQs, (Accessed on 08/15/2020).

[11] "What is business continuity? - business continuity 101," https://www.mha-it.com/2017/08/01/what-is-business-continuity/, (Accessed on 08/15/2020).

[12] "Intelligent storage device — san learning & clariion," http://storagesanbasic.blogspot.com/2012/05/intelligent-storage-device.html, (Accessed on 08/16/2020).

[13] "Module 4 - intelligent storage systems (iss) flashcards — quizlet," https://quizlet.com/215422126/module-4-intelligent-storage-systems-iss-flash-cards/, (Accessed on 08/16/2020).

[14] "What is storage provisioning? - definition from techopedia," https://www.techopedia.com/definition/16497/storage-provisioning, (Accessed on 08/16/2020).

[15] "What is data compression? — barracuda networks," https://www.barracuda.com/glossary/data-compression, (Accessed on 08/16/2020).

[16] "Redundant array of inexpensive disks (raid) - technical paper," https://www.howtoforge.com/redundant-array-of-inexpensive-disks-raid-technical-paper, (Accessed on 08/16/2020).

[17] "xtremio - google search," https://www.google.com/search?q=xtremio&rlz=1C1GCEB_enIE870IE870&sxsrf=ALeKk02Bs07QekwH8R8vGBKb125mX04olg:1597588927320&source=lnms&tbm=isch&sa=X&ved=2ahUKEwii7onE-p_rAhW9SRUIHYVfDaoQ_AUoAXoECBcQAw&biw=1280&bih=578#imgrc=ihlPaAR_30kfRM, (Accessed on 08/16/2020).

[18] D. Technologies, *Optimizing Storage Services for Applications with XtremIOX2*. Dell Technologies, 2017.

[19] "h15963-ss-isilon-all-flash.pdf," https://www.dellemc.com/resources/en-us/asset/data-sheets/products/storage/h15963-ss-isilon-all-flash.pdf, (Accessed on 08/17/2020).

[20] "Ecs overview and architecture," https://www.dellemc.com/fr-mg/collaterals/unauth/white-papers/products/storage-1/h14071-ecs-architectural-guide-wp.pdf, (Accessed on 08/17/2020).

[21] "Final version of nist cloud computing definition published — nist," https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published#:~:text=According%20to%20the%20official%20NIST,and%20released%20with%20minimal%20management, (Accessed on 08/18/2020).

[22] "History and vision of cloud computing — times of cloud," https://timesofcloud.com/cloud-tutorial/history-and-vision-of-cloud-computing/, (Accessed on 08/20/2020).

[23] "New tab," chrome://newtab/, (Accessed on 08/18/2020).

[24] "https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf," https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, (Accessed on 08/18/2020).

[25] "New tab," chrome://newtab/, (Accessed on 08/18/2020).

[26] "1: Cloud computing deployment models (mell and grance, 2011) — download scientific diagram," https://www.researchgate.net/figure/Cloud-Computing-Deployment-Models-Mell-and-Grance-2011_fig2_275036700, (Accessed on 08/18/2020).

[27] "Beware—this open database on google cloud 'exposes 200 million americans': Are you at risk?" https://www.forbes.com/sites/zakdoffman/2020/03/20/stunning-new-google-cloud-breach-hits-200-million-us-citizens-check-here-if-youre-now #4764005e8587, (Accessed on 08/18/2020).

[28] "Microsoft azure: A cheat sheet - techrepublic," https://www.techrepublic.com/article/microsoft-azure-the-smart-persons-guide/, (Accessed on 08/20/2020).

[29] "Amazon web services (aws): A cheat sheet - techrepublic," https://www.techrepublic.com/article/amazon-web-services-the-smart-persons-guide/, (Accessed on 08/20/2020).

[30] "Google cloud platform: A cheat sheet - techrepublic," https://www.techrepublic.com/article/google-cloud-platform-the-smart-persons-guide/, (Accessed on 08/20/2020).

[31] "Microsoft named a leader in gartner's public cloud storage services magic quadrant - mspoweruser," https://mspoweruser.com/microsoft-named-a-leader-in-gartners-public-cloud-storage-services-magic-quadrant/, (Accessed on 08/18/2020).

[32] "Resource access management in azure - cloud adoption framework — microsoft docs," https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/govern/resource-consistency/resource-access-management, (Accessed on 08/20/2020).

[33] "Storage account overview - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview, (Accessed on 08/20/2020).

[34] "Storage account overview - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview#types-of-storage-accounts, (Accessed on 08/19/2020).

[35] "Regions and availability zones in azure — microsoft docs," https://docs.microsoft.com/en-us/azure/availability-zones/az-overview, (Accessed on 08/20/2020).

[36] "Regions and availability zones in azure — microsoft docs," https://docs.microsoft.com/en-us/azure/availability-zones/az-overview#terminology, (Accessed on 08/19/2020).

[37] "Data redundancy - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/common/storage-redundancy#geo-zone-redundant-storage, (Accessed on 08/19/2020).

[38] "Introduction to blob (object) storage - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction, (Accessed on 08/19/2020).

[39] "Understanding block blobs, append blobs, and page blobs - azure storage — microsoft docs," https://docs.microsoft.com/en-us/rest/api/storageservices/understanding-block-blobs--append-blobs--and-page-blobs, (Accessed on 08/20/2020).

[40] "Hot, cool, and archive access tiers for blobs - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-storage-tiers?tabs=azure-portal, (Accessed on 08/20/2020).

[41] "Get started with azure queue storage using .net - azure storage — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/queues/storage-dotnet-how-to-use-queues?tabs=dotnet, (Accessed on 08/20/2020).

[42] "Introduction to azure files — microsoft docs," https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction, (Accessed on 08/20/2020).

[43] "Overview of azure table storage — microsoft docs," https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-overview, (Accessed on 08/19/2020).

[44] "Overview of azure table storage — microsoft docs," https://docs.microsoft.com/en-us/azure/cosmos-db/table-storage-overview, (Accessed on 08/20/2020).

[45] "Understanding the table service data model (rest api) - azure storage — microsoft docs," https://docs.microsoft.com/en-us/rest/api/storageservices/Understanding-the-Table-Service-Data-Model, (Accessed on 08/19/2020).

[46] "Understanding the table service data model (rest api) - azure storage — microsoft docs," https://docs.microsoft.com/en-us/rest/api/storageservices/Understanding-the-Table-Service-Data-Model, (Accessed on 08/19/2020).

[47] "Distribute data globally with azure cosmos db — microsoft docs," https://docs.microsoft.com/en-us/azure/cosmos-db/distribute-data-globally, (Accessed on 08/20/2020).

[48] "Distribute data globally with azure cosmos db — microsoft docs," https://docs.microsoft.com/en-us/azure/cosmos-db/distribute-data-globally, (Accessed on 08/19/2020).

[49] "Azure table storage support in azure cosmos db — microsoft docs," https://docs.microsoft.com/en-us/azure/cosmos-db/table-support, (Accessed on 08/19/2020).

[50] "Azure disk storage overview - azure virtual machines — microsoft docs," https://docs.microsoft.com/en-us/azure/virtual-machines/managed-disks-overview, (Accessed on 08/20/2020).

[51] "How to attach and mount vhd files to azure virtual machines," https://cloud.netapp.com/blog/azure-virtual-machines-attach-mount, (Accessed on 08/19/2020).

[52] "Azure disk storage overview - azure windows virtual machines — microsoft docs," https://docs.microsoft.com/en-us/azure/virtual-machines/windows/managed-disks-overview, (Accessed on 08/19/2020).

[53] "Azure disk storage overview - azure windows virtual machines — microsoft docs," https://docs.microsoft.com/en-us/azure/virtual-machines/windows/managed-disks-overview, (Accessed on 08/19/2020).

[54] “Azure managed vs unmanaged disks : The choice — buildwindows,” https://buildwindows.wordpress.com/2017/05/31/azure-managed-vs-unmanaged-disks-the-choice/, (Accessed on 08/19/2020).

[55] “Azure managed vs unmanaged disks : The choice — buildwindows,” https://buildwindows.wordpress.com/2017/05/31/azure-managed-vs-unmanaged-disks-the-choice/, (Accessed on 08/19/2020).

[56] “Azure managed vs unmanaged disks : The choice — buildwindows,” https://buildwindows.wordpress.com/2017/05/31/azure-managed-vs-unmanaged-disks-the-choice/, (Accessed on 08/19/2020).

[57] “Configure azure storage firewalls and virtual networks — microsoft docs,” https://docs.microsoft.com/en-us/azure/storage/common/storage-network-security, (Accessed on 08/19/2020).

[58] “Scalability and performance targets for standard storage accounts - azure storage — microsoft docs,” https://docs.microsoft.com/en-us/azure/storage/common/scalability-targets-standard-account, (Accessed on 08/19/2020).

[59] “Peer-to-peer - wikipedia,” https://en.wikipedia.org/wiki/Peer-to-peer, (Accessed on 08/20/2020).

[60] “Survey.pdf,” http://courses.cs.vt.edu/~cs5204/fall08-kafura/Papers/FileSystems/Survey.pdf, (Accessed on 08/20/2020).

[61] “Peer-to-peer [book],” https://www.oreilly.com/library/view/peer-to-peer/059600110X/, (Accessed on 08/20/2020).

[62] “Peer-to-peer [book],” https://www.oreilly.com/library/view/peer-to-peer/059600110X/, (Accessed on 08/20/2020).

[63] “mzpro2.pdf,” http://archive.cone.informatik.uni-freiburg.de/teaching/seminar/p2p-networks-w06/submissions/mzpro2.pdf, (Accessed on 08/20/2020).

[64] “Aol’s nullsoft creates software for swapping mp3s - cnet,” https://www.cnet.com/news/aols-nullsoft-creates-software-for-swapping-mp3s/, (Accessed on 08/20/2020).

[65] “Aol’s nullsoft creates software for swapping mp3s - cnet,” https://www.cnet.com/news/aols-nullsoft-creates-software-for-swapping-mp3s/, (Accessed on 08/20/2020).

[66] “Gnutellaprotocol0.pdf,” https://courses.cs.washington.edu/courses/cse522/05au/gnutella_protocol_0.4.pdf, (Accessed on 08/20/2020).

[67] “Instantly move money to all corners of the world — ripple,” https://ripple.com/, (Accessed on 08/20/2020).

[68] “Privacy-protecting digital currency — zcash,” https://z.cash/, (Accessed on 08/20/2020).

[69] “Hyperledger – open source blockchain technologies,” https://www.hyperledger.org/, (Accessed on 08/20/2020).

[70] "Multichain — open source blockchain platform," https://www.multichain.com/, (Accessed on 08/20/2020).

[71] "Home — quorum," https://www.goquorum.com/, (Accessed on 08/20/2020).

[72] "Tendermint," https://tendermint.com/, (Accessed on 08/20/2020).

[73] "Bitcoin - open source p2p money," https://bitcoin.org/en/, (Accessed on 08/20/2020).

[74] "Bitcoin - open source p2p money," https://bitcoin.org/en/, (Accessed on 08/20/2020).

[75] "A review on consensus algorithm of blockchain - ieee conference publication," https://ieeexplore.ieee.org/document/8123011, (Accessed on 08/20/2020).

[76] "Sybilguard: Defending against sybil attacks via social networks - ieee journals & magazine," https://ieeexplore.ieee.org/document/4542826, (Accessed on 08/20/2020).

[77] "Controlled supply - bitcoin wiki," https://en.bitcoin.it/wiki/Controlled_supply, (Accessed on 08/20/2020).

[78] "Digital signature - wikipedia," https://en.wikipedia.org/wiki/Digital_signature, (Accessed on 08/20/2020).

[79] "Public-key cryptography - wikipedia," https://en.wikipedia.org/wiki/Public-key_cryptography, (Accessed on 08/20/2020).

[80] "Ethereum: State of knowledge and research perspectives," https://www.researchgate.net/publication/323234138_Ethereum_State_of_Knowledge_and_Research_Perspectives, (Accessed on 08/20/2020).

[81] "Vitalik buterin - wikipedia," https://en.wikipedia.org/wiki/Vitalik_Buterin, (Accessed on 08/20/2020).

[82] "Launching the ether sale — ethereum foundation blog," https://blog.ethereum.org/2014/07/22/launching-the-ether-sale/, (Accessed on 08/20/2020).

[83] "Ethereum (eth) - all information about ethereum ico (token sale) - ico drops," https://icodrops.com/ethereum/, (Accessed on 08/20/2020).

[84] "proof-of-stake-faqs — ethereum wiki," https://eth.wiki/concepts/proof-of-stake-faqs, (Accessed on 08/20/2020).

[85] "What is ethereum 2.0? — consensys," https://consensys.net/blog/blockchain-explained/what-is-ethereum-2/, (Accessed on 08/20/2020).

[86] "Sharding on ethereum - ethhub," https://docs.ethhub.io/ethereum-roadmap/ethereum-2.0/sharding/, (Accessed on 08/20/2020).

[87] "(pdf) beigepaper: An ethereum technical specification — micah dameron - academia.edu," https://www.academia.edu/36597698/Beigepaper_An_Ethereum_Technical_Specification, (Accessed on 08/20/2020).

[88] "Ethereum yellow paper walkthrough (2/7)," https://www.lucassaldanha.com/ethereum-yellow-paper-walkthrough-2/, (Accessed on 08/20/2020).

[89] "paper.pdf," https://gavwood.com/paper.pdf, (Accessed on 08/20/2020).

[90] "Storj labs · github," https://github.com/Storj/, (Accessed on 08/20/2020).

[91] "Initial coin offering (ico) definition," https://www.investopedia.com/terms/i/initial-coin-offering-ico.asp, (Accessed on 08/20/2020).

[92] "Storj (storj) - all information about storj ico (token sale) - ico drops," https://icodrops.com/storj/, (Accessed on 08/20/2020).

[93] "storjv2.pdf," https://storj.io/storjv2.pdf, (Accessed on 08/20/2020).

[94] "Storj's decentralized cloud storage service 'tardigrade' goes live - coindesk," https://www.coindesk.com/storjs-decentralized-cloud-storage-service-tardigrade-goes-live, (Accessed on 08/20/2020).

[95] "Github - ethersphere/bee: Ethereum swarm bee," https://github.com/ethersphere/bee, (Accessed on 08/20/2020).

[96] "Swarm bee documentation," https://swarm-gateways.net/bzz:/docs.swarm.eth/, (Accessed on 08/20/2020).

[97] "Github - ethereum/devp2p: Ethereum peer-to-peer networking specifications," https://github.com/ethereum/devp2p, (Accessed on 08/20/2020).

[98] "libp2p," https://libp2p.io/, (Accessed on 08/20/2020).

[99] "2. architectural overview — swarm 0.5 documentation," https://swarm-guide.readthedocs.io/en/latest/architecture.html, (Accessed on 08/20/2020).

[100] "Sha-3. - wikipedia," https://en.wikipedia.org/wiki/SHA-3., (Accessed on 08/20/2020).

[101] "1. introduction — swarm 0.5 documentation," https://swarm-guide.readthedocs.io/en/latest/introduction.html, (Accessed on 08/20/2020).

[102] "Background," http://web.cs.ucla.edu/classes/cs217/05BitTorrent.pdf, (Accessed on 08/20/2020).

[103] "Git," https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control., (Accessed on 08/20/2020).

[104] "Content-addressable storage - wikipedia," https://en.wikipedia.org/wiki/Content-addressable_storage, (Accessed on 08/20/2020).

[105] "Tt.dvi," http://www-kiv.zcu.cz/~ledvina/DHT/tanner.pdf, (Accessed on 08/20/2020).

[106] "Tt.dvi," http://www-kiv.zcu.cz/~ledvina/DHT/tanner.pdf, (Accessed on 08/20/2020).

[107] “Tt.dvi,” http://www-kiv.zcu.cz/~ledvina/DHT/tanner.pdf, (Accessed on 08/20/2020).

[108] “Github - ipfs/go-ipfs: Ipfs implementation in go,” https://github.com/IPFS/go-IPFS, (Accessed on 08/20/2020).

[109] “Github - ipfs/go-ipfs: Ipfs implementation in go,” https://github.com/IPFS/go-IPFS.

[110] “Implementation status — ipfs docs,” https://docs.ipfs.io/project/implementation-status/, (Accessed on 08/20/2020).

[111] “https://ipfs.io/ipfs/qmr7gsqm93cx5eag6a6yrznde1fqv7ul6x1o4k7zrja3lx/ipfs.draft3.pdf.” https://ipfs.io/IPFS/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/IPFS.draft3.pdf., (Accessed on 08/20/2020).

[112] “Public gateway checker — ipfs,” https://ipfs.github.io/public-gateway-checker/, (Accessed on 08/20/2020).

[113] “Ipld - the data model of the content-addressable web,” https://ipld.io/, (Accessed on 08/20/2020).

[114] “Directed acyclic graph. - wikipedia,” https://en.wikipedia.org/wiki/Directed_acyclic_graph., (Accessed on 08/20/2020).

[115] “Ipns — ipfs docs,” https://docs.ipfs.io/concepts/ipns/, (Accessed on 08/20/2020).

[116] “Domain name system - wikipedia,” https://en.wikipedia.org/wiki/Domain_Name_System, (Accessed on 08/20/2020).

[117] “libp2p,” https://libp2p.io/, (Accessed on 08/20/2020).

[118] “download,” http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.4986&rep=rep1&type=pdf, (Accessed on 08/20/2020).

[119] “Github - multiformats/multiaddr: Composable and future-proof network addresses,” https://github.com/multiformats/multiaddr, (Accessed on 08/20/2020).

[120] “Dht-book.dvi,” https://pdfs.semanticscholar.org/8859/3bf30b9de1497ba22b1136ebc433f648bbc8.pdf., (Accessed on 08/20/2020).

[121] “https://ipfs.io/ipfs/qmr7gsqm93cx5eag6a6yrznde1fqv7ul6x1o4k7zrja3lx/ipfs.draft3.pdf,” https://ipfs.io/IPFS/QmR7GSQM93Cx5eAg6a6yRzNde1FQv7uL6X1o4k7zrJa3LX/IPFS.draft3.pdf, (Accessed on 08/20/2020).

[122] “Dag builder visualization,” https://dag.ipfs.io/, (Accessed on 08/20/2020).

[123] “Github - multiformats/cid: Self-describing content-addressed identifiers for distributed systems,” https://github.com/multiformats/cid, (Accessed on 08/20/2020).

[124] “Cid inspector — ipfs,” https://cid.ipfs.io/, (Accessed on 08/20/2020).

[125] “Cid inspector — ipfs,” https://cid.ipfs.io/, (Accessed on 08/20/2020).

[126] "Citeseerx — unknown file type," http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.6773&rep=rep1&type=pdf., (Accessed on 08/20/2020).

[127] "https://ipfs.io/ipfs/qmru1jj1knd9ftzjfwm4x9yta2wfxn1w2efk7mgtmj8xgk." https://ipfs.io/IPFS/QmRU1jJ1kNd9fTzjFwM4X9YtA2wfXN1W2eFK7mgTMJ8xgK., (Accessed on 08/20/2020).

[128] "filecoin.pdf," https://filecoin.io/filecoin.pdf, (Accessed on 08/20/2020).

[129] "proof-of-replication.pdf," https://filecoin.io/proof-of-replication.pdf, (Accessed on 08/20/2020).

[130] "What sets us apart: Filecoin's proof system," https://filecoin.io/blog/filecoin-proof-system/, (Accessed on 08/20/2020).

[131] "Github - filecoin-project/consensus: Filecoin consensus work," https://github.com/filecoin-project/consensus, (Accessed on 08/20/2020).

[132] "What is filecoin? — filecoin docs," https://docs.filecoin.io/introduction/what-is-filecoin/, (Accessed on 08/20/2020).

[133] "Exploration for graphsync · github," https://gist.github.com/vmx/fc89c8625b2e59dd4af3600ab7a98bda, (Accessed on 08/20/2020).

[134] "filecoin.pdf," https://filecoin.io/filecoin.pdf, (Accessed on 08/20/2020).

[135] "Connect to a network — filecoin docs," https://docs.filecoin.io/how-to/networks/, (Accessed on 08/20/2020).