**Durable Functions:**

A durable function is an extension of Microsoft Azure functions which are used to create a long lasting workflows with high quality performance and response time.

* Azure durable functions can be used to:

1. Maintain state of the app
2. Orchestrate workflows
3. Maximize serverless benefits

* Durable functions are used in complex serverless environments scenarios.
* Durable functions are capable of retaining states during their execution.

**Benefits of using durable functions:**

1. They can be used for writing the event driven code which can help in asynchronous concept in their implementation.
2. The durable functions can be chained together through patterns.
3. Durable functions be used in coordination or series on which they execute.
4. The state is managed in durable functions.

**Orchestration Functions:**

It is way to implement durable functions which allows to define stateless workflows.

**Benefits of orchestrations functions:**

1. Workflows can be defined in code rather than using JSON notation or any design tool.
2. Functions calls can be made in both ways asynchronously or synchronously.
3. Checkpoints in Azure handle functions called asynchronously when they are called, waited and recalled. This is done for making them cost effective.

There are three types durable functions:

1. **Client Functions:** These type if durable functions run when there is any type of event performed from client side and they act as an entry point as well.
2. **Orchestrator Functions:** These are the functions which define how logic is written in actions and all about their execution and order of execution can be written in C# or Javascript.
3. **Activity Functions:** These define the task performed by the durable functions or orchestrator functions.

**Application patterns of durable function:**

1. Functions chaining
2. Fam in / Fam out
3. Async HTTP apps
4. Monitor
5. Human interaction

* A durable functions provides the infrastructure to create long running workflows without requiring managing the state manually.
* Time concerns can be addressed through following on durable functions:

1. Timeouts
2. Escalation paths

* Durable provides the time outs which can be helpful in asynchronous actions of function. They provide delays in execution of function.
* Azure functions core tools are used for creating Azure functions locally.
* These are the set of commands line tools which can be used to develop the Azure functions in the local environment and then publish on Azure.
* Functions hosts are hosts which do following things:

1. Power functions.
2. Handles configuration, listens to events, triggers and more.
3. Used for waiting log outputs
4. It runs automatically when function app starts.

**Webhooks:**

It is a source for trigging a function when an action is happened or when an event is performed.

* A webhook can trigger a function when an HTTP request happens.
* Webhooks are under-defined HTTP call backs through which a function is called.
* Webhooks can be setup in GitHub on any type repositories which can be personal or organizational repositories which can trigger a function when an event is occurred.
* Webhooks can be built through GitHub and managed by webhooks API.
* Payload is URL of server which will receive the webhook post requests.
* Webhooks can be delivered through two content types:

1. Application / JSON type
2. Application / x-www-from-urlencoded

* For verification and security purpose there should be secret key setup provided by webhooks to limit and validate the request sent from github.
* This above process is ensured by a Hash signature which is used for verification and also sent with all the event requests occurring.
* This hash key is generated by GitHub when an event occurs.

**Azure Functions for automatic updates in a webpages:**

* This relates to the creation of function which is triggered by the change in the web page data and set data of web page to newly updated date.
* The function can be set to update the page on a time which is set inside it as an internal.
* The function can be made without setting intervals as they can be triggered as data is changed in the database or the source, from which data is coming.
* These connections are made through bindings.
* These actions can also be achieved through persistent connections which allows server to push data to client side as it is changed.
* This connection is made by Signal Rs.
* The signal R is a combination of technologies which helps in managing changing on server side and then moving them on the client side.
* This is simply used to manage connections between server and client side.
* It uses a way named as “Transport” to look for API’s which will transfer the data as well as it gets those API’s which are supported by client side.

e.g:

* For html client side webhooks API is used.
* If above client side does not supports webhooks API then it is turned back to eventSource which is server sent events.
* For older sides ajax long polling or forever frame is used.

**Benefits of signal R service:**

* It gives you future-proofing through which application works well although it is updated. You need only specify the version of service you want.
* It gracefully degrades the application as per demand of API and need of application.

**Building Azure management API:**

* Azure management can be used to create an stable api of micro-services which are form of azure functions that contains the logic for different tasks.
* These APIs can be created from serverless Architecture by managing micro-services.
* Azure API management is a cloud service used to make, secure, change or update APIs.

**Benefits of using APIM:**

* Easier to deliver micro-services.
* Bugs and errors are reduced.
* It allows high security and cost efficient deployment and solution.
* It supports new and imported from other services.

**Connect your services with Microsoft Azure service Bus.**

**Queue Storage:**

Azure queue storage is a storage used to store messages which can be accessed by any REST API interface.

* The message contains the raw data to another service. The service is sending is called as the sending service and service which receives the sent messages is called as destination service or component.
* Queues use topics which act as a bridge between receiver and sender.
* Through topics one message can be sent to multiple receivers.
* Sent messages through topics can be sent to receivers based on topics.

**Benefits of using Queues:**

1. Increased reliability
2. Message delivery guarantees
3. Transactional support

**How to choose communication method:**

* If communication is done using events consider using event grid or event hub.
* If single message is delivered to many instances or places choose service bus or Queue storage.

**Azure Queue Services:**

It is a cloud based service which implements Queue storage for the apps deployed on azure platform.

* Apps can access the queues using the REAT API.
* The size of Queue in Microsoft Azure depends upon its demand, lower the demand smaller will be the queue, higher the demand larger will be the queue.
* An azure queue can be accessed by three main ways:

1. Storage account name
2. Queue name
3. Authentication token or key

**Azure Event Hub:**

An azure event hub is cloud based service which is used to receive event requests and process the actions on the events using event answering components or services.

e.g: These events can be HTTP requests used for any action.

* The data which is send to event hub is known as publishing data, is sent by publisher.
* The receiver of the data from event hub is known as subscriber.

**Publisher:**

The publisher can be any app or device which sends the data about apps.

* It can be done by two things:

1. AMQP => For frequent sending of data
2. HTTPS => For intermediate publishing

* Event hubs can have consumer groups which uses multiple subscribers to publish the events. These subscribers can be individual for publishers.
* The two main steps to create and configure the new event hubs:

1. Define event hubs namespace
2. Create an event hub

**Namespace for Event Hubs:**

It is an entity used for managing one or more event hubs.

* It involves the following configurations:

1. Define namespace settings:

* Certain settings are defined at the namespace level.
* Apply to all the event hubs with in the namespace.
* Default values of 1 for capacity and standard pricing tier.

1. Optional properties:

* Enable kafka.
* Make namespace zone redundant.
* Auto-inflate and auto-inflate maximum through out units.

**Command used to create namespace:**

**az eventhubs create –names $NS\_NAME**

**Creating an event hub:**

Following are the requirements to create an event hub:

1. Event hub name:

* Unique name.
* 1 to 50 characters long.
* Only letters, numbers, periods, hyphens and underscores.
* Starts and ends with a letter or number.

1. Portion court:

* Required partitions (2-32).
* Directly related to expected concurrent consumers.
* Separates the messages stream.
* If not defined, defaults to 4.

1. Message retention:

* Number of days message remains available .
* Between 1to7.
* Default value is 7.

Event hub can be configured to stream data to:

1. Azure blob storage.
2. Azure data lake.

**CLI Command:**

1. **az event hub create –name $HUBNAME –namespace –name $ NS\_NAME**
2. **az event hub show –name $HUBNAME –namespace –name $ NS\_NAME**

* Event hubs offer auto reconnecting as the event hub is shut down and connects automatically to application as service is available. The event received by event hub are not lost in this and successfully transferred to app as it is available.

**Azure Event Grid:**

It is a cloud based service used to manage event senders and event receivers.

1. It is simple to use event grids in azure and can easily be connected to azure event hubs.
2. It is also beneficial in connectivity with multiple subscribers to a publishers.
3. It is also beneficial in filtering events sent by publisher to be delivered to subscribers.
4. It is reliable.
5. Its throughput is high which accepts millions requests per second.
6. It has built events which starts it quickly.
7. It also supports custom events.

* The data storage in azure cloud is decided by the types of data which is to be stored in database.
* Key factors in choosing best data storage are:

1. What type of data is in action?
2. What are the ways or task in which data is going to be used?
3. How it is going to benefit your solution or business model?

**Structured Data:**

A data which is stored under a strict schema is known as structured data.

* This type of data can easily be manipulated by SQL.

**Semi Structured Data:**

A type of data which contains some fields as structured and some fields as unstructured data.

* This semi structured data can operate under SQL commands.
* The semi Structured data is determined in form of JSON, XML and YAML file systems. These are the methods to determine data and use different types of languages and forms to perform this.

**Unstructured Data:**

Unstructured data is a form of data which is neither defined in tables or file systems. It is only analyzed for the use of business logic and work. This can be shown in the form of Graphs.

* Questions that should be thought in mind as we work with any type of data:

1. Simple lookups.
2. Query database.
3. Action updates.
4. Analytical queries.
5. Quick turnaround.

**Transaction:**

A transaction is a logic group of actions which execute together.

* This approach is used in a data logic when change in one type of data changes another type of data.
* Transactions follow rule of ACID which refers to:

A → Atomicity

C → Consistency

I → Isolation

D → Durability

**Atomicity:**

* It ensures that transaction happens once at a time.
* Its operations should be interdependent.

**Consistency:**

* It ensures that data is consistent before and after the transaction.

**Isolation:**

* It ensures the one transaction is not affected by another transaction.

**Durability:**

* Changes made due to transaction should be permanently saved into the system.
* Changes made should be committed.

OLTP → Online Transaction Processing

OLAP → Online Analytical Processing

* The first one supports store and records transactions.
* The second one supports complete analytics.
* The first type systems supports more amount of users and handle large amount of data, but these are used to handle simple transactions which take less tome to complete.
* OLAP is used for complex analysis like business analytics and complex data tactics and statistics.
* The data solution should be chosen on following demands.

→ Data classification.

→ Operations.

→ Latency and Throughput.

→ Transactional supports and needs.

**Azure Storage Account:**

A storage account is a container that combines azure storage services together.

It can only include data storage from Azure services.

Settings that storage account controls are:

* Subscription
* Location
* Performance
* Replication
* Access Tier
* Secure Transfer
* Virtual Networks

An Azure subscription can hold up to 200 Storage Accounts.

Each Storage Account can hold up to 5 TB of data stored in it.

The data types Azure Storage support:

* Blobs
* Files
* Queues
* Table Storage

The most used storage account in Azure is standard general purpose V2 and another account is premium block blobs and this is only for page blobs.

Azure provides many types of REST API’s to perform storage operation through internet.

To work with the data stored in the storage account wr require two things:

* Access key
* REST API endpoint

**Azure Security Services:**

Key things for an administrator for the data stored in Azure are:

1. Protect data at rest.

2. Protect data in transit

3. Support browser cross domain access.

4. Control access to data

5. Audit storage access

**Data Encryption in Azure:**

All the data written in MS Azure is encrypted by service Encryption or SSE with 256 bits AES cipher. This data is automatically encrypted in Azure by SSE as it is written.

As data is used for reading in Azure it automatically decrypted by SSE and this process also not effects performance.

**Virtual Machine Encryption:**

Azure gives the facility by Azure Disk Encryption for VM’s encryption.

This an encryption is OS specific.

**Azure Key Vault:**

The stores key access including VM disk encryption. This ensures the safety of data in a manner that if some gets access to VM’s data it still can not access the data.

For secure communication between client and server it is necessary to use HTTPS request methods.

An secure way to access the data is a role based access of data. This is highly secure and can resist unauthorized persons from access the data.

Storage keys are the most important asset for an storage account because they ate responsible for the user interaction with storage account. It is necessary to regenerate storage keys if an unwanted situation occurs.

**Shared Access Signatures:**

It is string like token used to authenticate the third partly application to use the storage account.

It defines permissions and constrains for any client application which is interacting with Azure storage account data.

There are two types of SAS:

**1. Service level SAS:**

This SAS is used for authentication of specific resource in the storage account.

**2. Account level SAS:**

This type of SAS is used to allow the service level SAS resources plus some additional ones as well.

It also authorizes user to create a file system in the STA.

**Azure Defender for Storage:**

It is an extra layer of security in Azure data storage which handles threats and triggers alerts as they occur.

It comes with its own pricing.

It is supported by blobs, Azure files and Azure data lake storage Gen 2.

It is supported by account types like Azure storage account V2, block blobs and blob storage.

It is mostly available to us and its nearly clouds.