### Questions

1. **Difference between design and architecture**
   * Architecture (**Skeleton or Blueprint**) faces towards strategy, structure and purpose, towards the abstract.
   * Design faces towards **implementation, DS- Algo etc** and practice, towards the concrete.
2. **Microservices Design Patterns (Domain based and Event based)**

* To begin with, in an **event-driven microservice** architecture, services communicate each-other via event messages. When business events occur, producers publish them with messages. At the same time, other services consume them through event listeners.
* Domain-Driven Design (**DDD**) is a software development approach around solving complex domain model; the solution revolves around the business model by connecting the implementation to the core business concepts. The common terminology between the business/domain experts and the development team are domain logic, subdomains, bounded contexts, context maps, domain models, and ubiquitous language as a way to collaborate and improve the application model and resolve any emerging domain-related issues.
* [microservices](https://codeburst.io/software-architecture-the-difference-between-architecture-and-design-7936abdd5830)

1. **SOLID** refers to **Single Responsibility, Open Closed, Liskov substitution, Interface Segregation and Dependency Inversion Principles**.
2. **Advantage and Disadvantage of Microservices** : ([Link](https://cloudacademy.com/blog/microservices-architecture-challenge-advantage-drawback/))

**Advantages**

* Improve fault Isolation
* Eliminate vendor or technology lock-in
* Smaller and faster deployments
* Scalability

**Disadvantages**

* Communication between services is complex
* More services equals more resources
* Global testing is difficult

1. **NoSQL vs RelationalDb**  ([Relational Vs NoSql Db](https://www.mongodb.com/scale/nosql-vs-relational-databases))

* NoSQL tends to be a better option for modern applications that have more complex, constantly changing data sets, requiring a flexible data model that doesn’t need to be immediately defined. Most developers or organizations that prefer NoSQL databases, are attracted to the agile features that allow them to go to market faster, make updates faster. Unlike traditional, SQL based, relational databases, NoSQL databases can store and process data in real-time.
* While SQL databases do still have some specific use cases, NoSQL databases have many features that SQL databases are not capable of handling without tremendous costs, and critical sacrifices of speed, agility, etc.

#### **About various microservices concepts**

- **Microservices** are a way of breaking large software projects into loosely coupled modules, which communicate with each other through simple Application Programming Interfaces (APIs).

- **API gateway** (An API gateway is an API management tool that sits between a client and a collection of backend services.)

- **Fault tolerance** (Fault tolerance is the property that enables a system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.)

- **Service discovery** (Service discovery is the process of automatically detecting devices and services on a network. Service discovery protocol (SDP) is a networking standard that accomplishes detection of networks by identifying resources. Traditionally, service discovery helps reduce configuration efforts by users who are presented with compatible resources, such as a bluetooth-enabled printer or server.)

- **Circuit breakers** ([Circuit Breaker](https://docs.microsoft.com/en-us/azure/architecture/patterns/circuit-breaker). Too many failed requests can cause a bottleneck, as pending requests accumulate in the queue. These blocked requests might hold critical system resources such as memory, threads, database connections, and so on, which can cause cascading failures. The Circuit Breaker pattern can prevent a service from repeatedly trying an operation that is likely to fail.)

- **Scaling of Microservice**s ([Scaling](https://dzone.com/articles/scaling-microservices-the-challenges-and-solutions#:~:text=Scalability%20is%20determined%20by%20how,able%20to%20process%20these%20tasks.&text=In%20order%20to%20scale%20successfully,part%20of%20a%20larger%20system.) : Maintaining Performance, Resource allocation )

**- Security management of microservices** ([Security](https://techbeacon.com/app-dev-testing/8-best-practices-microservices-app-sec) : Use OAuth Routes, Use Crypto Code (NaCl or Libsodium, Use security update feature, Get container out in public network and Scanning of the container)

- **Communication between microservices** (**Microsoft** : [Communication Microservices](https://docs.microsoft.com/en-us/azure/architecture/microservices/design/interservice-communication)) (

Synchronous communication. In this pattern, a service calls an API that another service exposes, using a protocol such as HTTP or gRPC. This option is a synchronous messaging pattern because the caller waits for a response from the receiver.

Asynchronous message passing. In this pattern, a service sends a message without waiting for a response, and one or more services process the message asynchronously.)

- Coordination/communication issues and other issues related to microservices

- Orchestration and monitoring of microservices

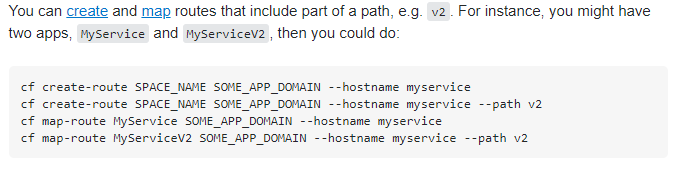
([for both above answer](https://www.idexcel.com/blog/microservices-architecture-advantages-and-challenges/#:~:text=Challenges%20of%20MicroService&text=Inter%20Service%20Communication%20%E2%80%93%20MicroServices%20will,framed%20using%20HTTP%2FESB%20etc.&text=Health%20Monitoring%20%E2%80%93%20There%20are%20more,developed%20using%20different%20programming%20languages.))

- The **service mesh** acts as a proxy that intercepts network communication between microservices in the cluster. Currently, the service mesh concept applies mainly to container orchestrators, rather than serverless architectures.

### Example questions

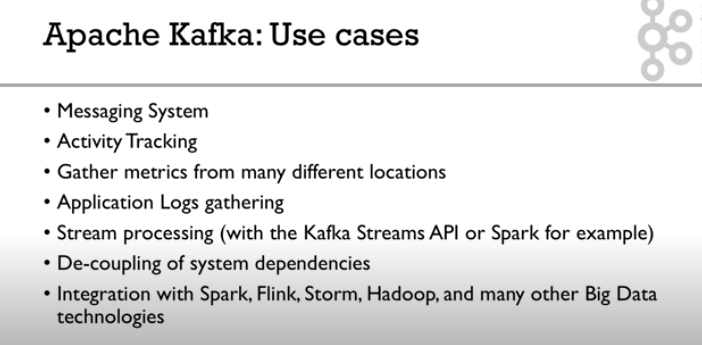
* Some quick questions about the manageability of a microservices deployment:
* How do we start and stop a fleet of services? ([Kubernetes instead Fleet](https://coreos.com/fleet/docs/latest/launching-containers-fleet.html))
* How do we aggregate **logs**/metrics/SLAs across microservices? (Use a **centralized logging service or Cloud watch**  that aggregates logs from each service instance. The users can search and analyze the logs. They can configure alerts that are triggered when certain messages appear in the logs.)
* How do we discover services in an elastic environment where they can be coming, going, moving, etc.? ([service discovery](https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/))
* How do we do **Load balancing**? (Load balancing is defined as the methodical and efficient distribution of network or application traffic across multiple servers in a server farm. Each load balancer sits between client devices and backend servers, receiving and then distributing incoming requests to any available server capable of fulfilling them.

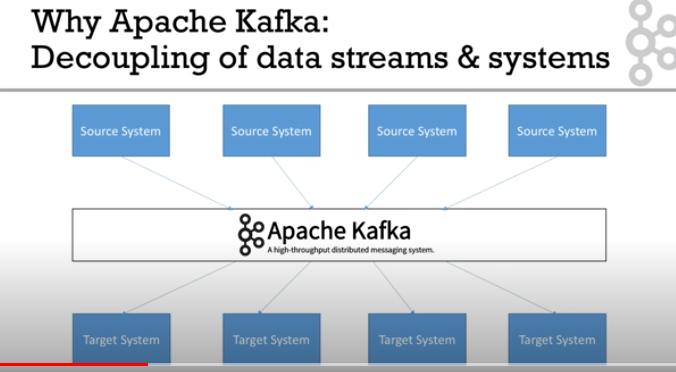
-- **Kubernetes** using kube-proxy which manages virtual IP used by Service.)

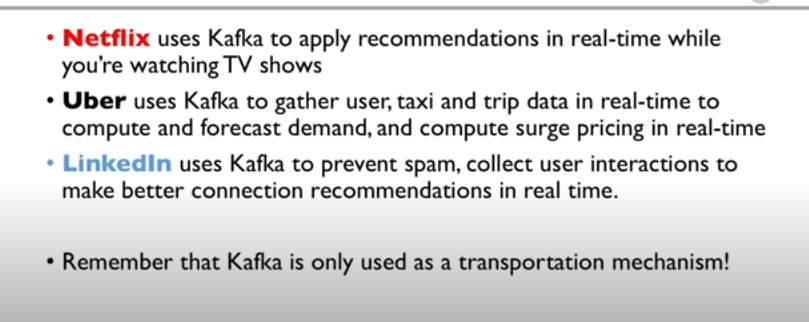
* How do we learn about the health of our cluster or individual services? (One approach to monitoring all cluster nodes is to create a special kind of Kubernetes pod called [DaemonSets](https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/). Kubernetes ensures that every node created has a copy of the DaemonSet pod, which virtually enables one deployment to watch each machine in the cluster)
* How do we restart services that have fallen over? (kubectl run -it --rm --restart=Never alpine --image=alpine sh)
* How do we do fine-grained API routing? [Fine grained API](https://horizontal.blog/2020/06/09/fine-grained-vs-coarse-grained-apis/) (Fine-grained APIs are designed to easily scale and improve the performance of your APIs)
* How do we secure our services?
* How do we throttle or disconnect parts of a cluster if it starts to crash or act unexpectedly? ( -Elastic Search - An object store where all logs are stored. )
* How do we deploy multiple versions of a service and route to them appropriately?
* 
* How do we make configuration changes across a large fleet of services? (YAML File in Kubernetes)

### System designs

* How we design apache kafka your own microservices [Kafka](https://developer.okta.com/blog/2020/01/22/kafka-microservices)
* Design for table booking app
* Microservices design for food app
* Photos uploader app
* While explaining the design we need to explain each component, interaction between components/services, where async call / where sync calls, how various events management between the services
* Various approaches for Migration of project to new teams and its challenges







### Coding Interview Question

//String line1 = "he,hello,the,there";

//String line2 = "hellothere";

Solution: hello there

11. Given an array [1,2,3,4,5] group them as possible sums like [((1,4), (2,3)), ((4,2),(1,5))]

### My Questions

(Java Solution already shared with Team)

1. convert digital watch to binary watch

2. find first duplicate character in the string

3. Find whether 2 strings are anagrams :

Fried --- fired

Gainly – laying

Sadder – dreads

4. Suppose there are 2 strings source =”abcde” target =”ab” . Your program should return true if target is there in source. And also if the reverse of target is there in source

5. An array has numbers from 1 to 20 . some of the numbers are missing if we sort the numbers in ascending order. Find out the missing numbers.

6. Find the first non repeating character in a string eg: helloworld h is the answer.

7. Find the palindrome permutations for Malayalam.

8. Give an array of anagrams [“Fried”, “Gainly” ,”fired”, “dreads” ,”laying”, “Sadder” ,”god”, “goat”]

Group them as follows:

[

[“Fried”, ”fired”],

[“Gainly”, ”laying”],

[“dreads”, “Sadder”],

[“god”],

[“goat”]

]

9. Find out all palindromic substrings from a give string eg: aabbaacabcadadad

10. Problem Statement:

Line1 is a dictionary of words. Convert Line 2 to a meaningful sentence by referring to the words in Line1.

Example1:

// String line1 = "I,A,AM,HE,HERE";

// String line2 = "IAMHERE";

Solution :I AM HERE

[https://www.geeksforgeeks.org/word-break-problem-using-backtracking/?ref=lbp](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fword-break-problem-using-backtracking%2F%3Fref%3Dlbp&data=02%7C01%7Calok.tripathi3%40wipro.com%7C4dbe7c83dc8e4541beb808d8592dbeab%7C258ac4e4146a411e9dc879a9e12fd6da%7C1%7C0%7C637357400295960750&sdata=XK%2BqS2uYTWypMhqRaXrWgrtmQsAO6s45dBBKpxqgOPA%3D&reserved=0)

[https://www.geeksforgeeks.org/word-break-problem-trie-solution/?ref=lbp](https://apc01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.geeksforgeeks.org%2Fword-break-problem-trie-solution%2F%3Fref%3Dlbp&data=02%7C01%7Calok.tripathi3%40wipro.com%7C4dbe7c83dc8e4541beb808d8592dbeab%7C258ac4e4146a411e9dc879a9e12fd6da%7C1%7C0%7C637357400295970744&sdata=Z9cRnu97s63bO%2FhkHwIfFGbBNlMIljT0B%2BboLm0dn44%3D&reserved=0)

Example2: