1.

1.
Answer in brief: **[2 * 3 = 6]**

1.
Explain with an example, relationship between microservices, Docker and kubernetes.

Microservices, Docker and Kubernetes are all related technologies that are used together to create a more efficient and agile approach to application development and deployment.

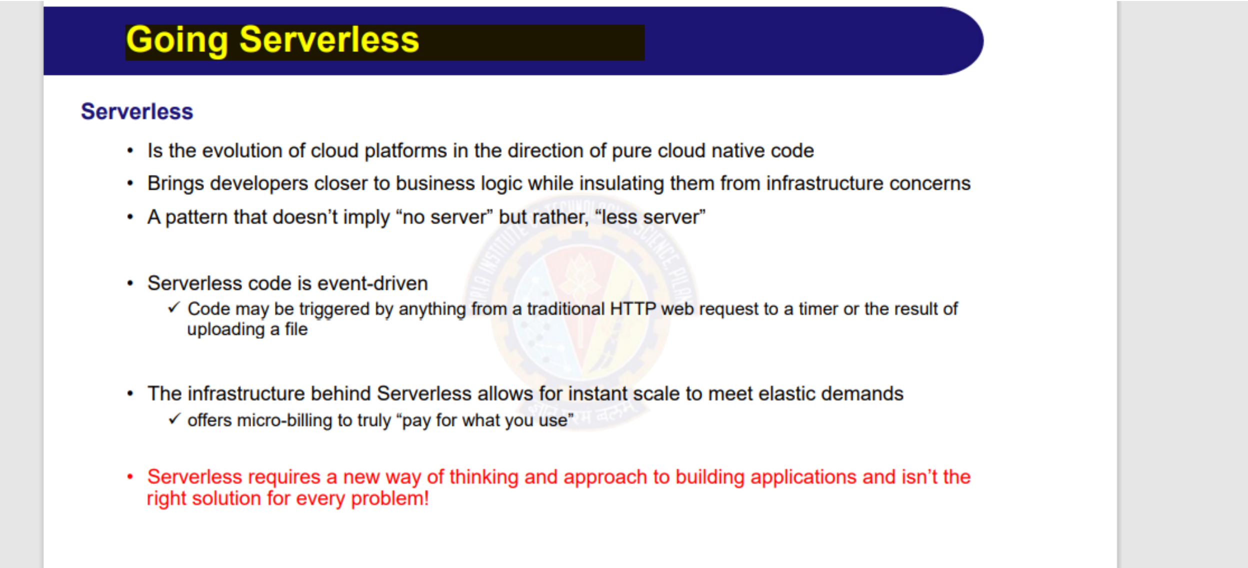
An example of the relationship between microservices, Docker and Kubernetes would be a web application. A web application consists of several different components, such as a database, web server, user authentication and so on. Each of these components can be broken down into smaller services that can be individually managed and deployed using Docker containers. Kubernetes can then be used to orchestrate and manage the containers, allowing for more efficient deployment and scalability of the application. By breaking down the application into smaller services, each with its own container, the application can be deployed and managed more efficiently, making it easier to update and maintain.

2. Consider a case of composite pattern - where an application fetches data from different APIs. For example, a dashboard that retrieves the required data from different sources such as logging services APIs, cloud based backends for consumption numbers, third-party analytics services to capture end-user interactions etc. Which type of API design style will be suitable for this scenario? Justify?

In this scenario, a hybrid API design style would be suitable. A hybrid API design style combines elements of the RESTful and RPC-style APIs to provide a unified interface for applications. It allows for a more flexible approach to API integration, as it provides a common interface for different data sources, making it easier for the application to access data from multiple sources. Additionally, it allows for a more granular approach to data access, allowing for more fine-grained control over what data is retrieved from each source.

3. Serverless computing can be thought of as no servers instead of less servers. Justify/Invalidate.

Invalidate. Serverless computing does not mean that there are no servers involved. Rather, it is a cloud computing execution model in which the cloud provider dynamically manages the allocation of machine resources. Serverless computing relies on servers to run code, but abstracts away the need for developers to manage the underlying infrastructure.



Going Serverless

Serverless

- Is the evolution of cloud platforms in the direction of pure cloud native code
- Brings developers closer to business logic while insulating them from infrastructure concerns
- A pattern that doesn't imply "no server" but rather, "less server"
- Serverless code is event-driven
  - ✓ Code may be triggered by anything from a traditional HTTP web request to a timer or the result of uploading a file
- The infrastructure behind Serverless allows for instant scale to meet elastic demands
  - ✓ offers micro-billing to truly "pay for what you use"
- Serverless requires a new way of thinking and approach to building applications and isn't the right solution for every problem!

FAAS

2.

Rishikesh is an enthusiastic traveler. When travelling he uses his DSLR camera a lot to capture the pictures of the surroundings. Also he uses these pictures in blogposts which narrates his journeys and experiences of the places which he has visited in the past. As his blogs are very informative, many readers find them quite useful when they plan their journeys to those places, hence many feedbacks are also shared by the readers.

Answer the following sub-questions based on the above narrative: **[2 + 2 + 2 = 6]**

1. What type of data is captured in this narrative?

2. Whether a file based system will be appropriate choice for such type of data? Why?

3. What other type of data storage will be suitable for such type of data? Justify with example

(a) The type of data captured in this narrative is textual, photographic and feedback data. -

(b) A file based system is not an appropriate choice for such type of data. A file based system is suited for storing structured data, while the data mentioned in the narrative is unstructured. It is difficult to store and retrieve unstructured data in a file based system.

(c) A database system is a more suitable choice for such type of data. A database system can store and retrieve unstructured data easily. For example, a database system can store Rishikesh's pictures, blog posts, and reader feedbacks in separate tables. It can later be used to extract data relevant to a particular journey, such as pictures and feedbacks.

.

3. Let’s assume that you need to design a mobile app for English language dictionary. It should serve as English language learning tool and provide word games built for every level of learner.

With trusted definitions and synonyms plus word puzzles, language quizzes, and spelling quizzes, this English dictionary and thesaurus app for Android should be optimized with your mobile device in mind to help you learn English or improve your English vocabulary. In addition to the trusted reference content from Dictionary.com and Thesaurus.com, this education app should include:

- Word Puzzle ► To Improve your vocabulary with fun spelling quizzes and vocabulary challenges.
- Word of the Day ► Learn a new word each day and expand your vocabulary education.

- Synonyms ► Get thesaurus content alongside your dictionary definitions.
- Audio pronunciations ► Never mispronounce another word.
- Voice search ► Find the definitions you're looking for anywhere, anytime. The app even offers up English spelling help. Not sure how a word is spelled? Say it out loud, and this app will find it for you.
- Grammar help ► Get grammar tips, word usage, and more to improve your writing.
- Favorite words and search history ► Customize your recently searched word list, and never forget the newest words you've learned
- Learner's dictionary ► Includes extra information about word usage for English learners

Answer the following sub questions based on this scenario: **[2 + 2 + 1 + 2 + 2 + 1 = 10]**

1. What will be the type of mobile dictionary application? Justify briefly.
2. What features of the mobile phone will be leveraged by this dictionary application?
3. What framework and programming language will be suitable for development of this type of application?
4. If this application needs to work in offline manner, then using a file for storing dictionary data will be an appropriate choice? Justify.
5. Every day when user first time opens up the application, a new "word of day" needs to be shown to him. Describe the factors that you will consider while designing this feature?
6. Whether using any Backend-as-a-Service (BaaS) for dictionary app will make any difference?

4. For the below mentioned cases, identify a suitable architectural style (discussed in the class) and provide an architectural block diagram and short description narrating the request-response flow between various components involved. **[8]**

1. Yours is a unified payment interface that enables transfer of money from one back account to another account and also has plans in mind to extend it for transfer of money between bank account and credit cards.

2. Yours is credit score management system that tracks the loans taken by the customer and updates the credit score on regular basis when an EMI is paid by the customer

3. Yours is a business that wants to adapt mobile-only application for supporting the business transactions and does not want to take headache associated with management and maintenance of infrastructure required for the application

4. You quickly need to build a prototype of the product before embarking on a more ambitious project, its less complex in nature and the team has expertise into conventional development and deployment approaches

****

[Yesterday 10:03 PM] NISHITH .

1. Data Source: The first factor to consider when designing this feature is the data source. It is important to identify the data source that will provide new words each day. This could be an online dictionary, an online thesaurus, or a database of words. 2. Word Difficulty: The second factor to consider is the difficulty of the word. Depending on the target audience, the word difficulty should be tailored accordingly. For example, if the target audience is children, the word difficulty should not be too high. 3. Word Relevance: The third factor to consider is the relevance of the word to the target audience. The word should be relevant to the target audience’s interests, needs, and/or experience. 4. Word Variety: The fourth factor to consider is the variety of words. In order to keep the feature engaging, it is important to provide a variety of words from different topics. 5. Word Frequency: The fifth factor to consider is the frequency of the words. It is important to ensure that the same words are not repeated too often. 6. Word Support: The sixth factor to consider is the support for the words. It is important to provide