1. **What are the Drawbacks of traditional monolithic applications?**
   * The large monolithic code base intimidates developers, especially ones who are new to the team. The application can be difficult to understand and modify. As a result, development typically slows down. Also, because there are not hard module boundaries, modularity breaks down over time. Moreover, because it can be difficult to understand how to correctly implement a change the quality of the code declines over time. It’s a downwards spiral.
   * **Overloaded IDE** – the larger the code base the slower the IDE and the less productive developers are.
   * **Overloaded web container** - the larger the application the longer it takes to start up. This had a huge impact on developer productivity because of time wasted waiting for the container to start. It also impacts deployment too.
   * **Continuous deployment is difficult** - a large monolithic application is also an obstacle to frequent deployments. In order to update one component, you have to redeploy the entire application. This will interrupt background tasks (e.g., Quartz job in a Java application), regardless of whether they are impacted by the change, and possibly cause problems. There is also the chance that components that haven’t been updated will fail to start correctly. As a result, the risk associated with redeployment increases, which discourages frequent updates. This is especially a problem for user interface developers, since they usually need to iterative rapidly and redeploy frequently.
   * **Scaling the application can be difficult** - a monolithic architecture is that it can only scale in one dimension. On the one hand, it can scale with an increasing transaction volume by running more copies of the application. Some clouds can even adjust the number of instances dynamically based on load. But on the other hand, this architecture can’t scale with an increasing data volume. Each copy of application instance will access all of the data, which makes caching less effective and increases memory consumption and I/O traffic.
   * **Obstacle to scaling development** - A monolithic application is also an obstacle to scaling development. Once the application gets to a certain size its useful to divide up the engineering organization into teams that focus on specific functional areas. For example, we might want to have the UI team, accounting team, inventory team, etc. The trouble with a monolithic application is that it prevents the teams from working independently. The teams must coordinate their development efforts and redeployments. It is much more difficult for a team to make a change and update production.
   * **Requires a long-term commitment to a technology stack** - a monolithic architecture forces you to be married to the technology stack (and in some cases, to a particular version of that technology) you chose at the start of development. With a monolithic application, can be difficult to incrementally adopt a newer technology. For example, let’s imagine that you chose the JVM.
2. **What is SAAS/PAAS/IAAS? (Definitions and Advantages)**
   * **Software as a Service (SAAS):**
   * **Definition:** This service provides the capabilities to the consumer to use the applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, like an internet browser (e.g., web-based email), or a program interface. The buyer or consumer doesn't manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or individual application capabilities, with the possible exception of limited user-specific application configuration settings. Software Adoption is fast and Easy.
   * **Advantages:**
   * **Gain access to sophisticated applications:** To provide SaaS apps to users, you don’t need to purchase, install, update or maintain any hardware, middleware or software. SaaS makes even sophisticated enterprise applications, affordable for organizations that lack the resources to buy, deploy and manage the required infrastructure and software themselves.
   * **Pay only for what you use:** You also save money because the SaaS service automatically scales up and down according to the level of usage.
   * **Use free client software:** Users can run most SaaS apps directly from their web browser without needing to download and install any software, although some apps require plugins. Y0ou don’t need to purchase and install special software for your users.
   * **Mobilize your workforce easily:** SaaS makes it easy to “mobilize” your workforce because users can access SaaS apps and data from any Internet-connected computer or mobile device. You don’t need to worry about developing apps to run on different types of computers and devices because the service provider has already done so. In addition, you don’t need to bring special expertise onboard to manage the security issues inherent in mobile computing. A carefully chosen service provider will ensure the security of your data, regardless of the type of device consuming it.
   * **Access app data from anywhere:** With data stored in the cloud, users can access their information from any Internet-connected computer or mobile device. And when app data is stored in the cloud, no data is lost if a user’s computer or device fails.
   * **Platform as a Service (PAAS):**
   * **Definition:** This service provides to the buyer or consumer to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The buyer doesn't 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.
   * **Advantages:**
   * By delivering infrastructure as a service, PaaS offers the same advantages as IaaS. But its additional features—middleware, development tools and other business tools—give you more advantages:
   * **Cut coding time:** PaaS development tools can cut the time it takes to code new apps with pre-coded application components built into the platform, such as workflow, directory services, security features, search and so on.
   * **Add development capabilities without adding staff:** Platform as a Service components can give your development team new capabilities without your needing to add staff having the required skills.
   * **Develop for multiple platforms—including mobile—more easily:** Some service providers give you development options for multiple platforms, such as computers, mobile devices and browsers making cross-platform apps quicker and easier to develop.
   * **Use sophisticated tools affordably:** A pay-as-you-go model makes it possible for individuals or organizations to use sophisticated development software and business intelligence and analytics tools that they could not afford to purchase outright.
   * **Support geographically distributed development teams:** Because the development environment is accessed over the Internet, development teams can work together on projects even when team members are in remote locations.
   * **Efficiently manage the application lifecycle:** PaaS provides all of the capabilities that you need to support the complete web application lifecycle: building, testing, deploying, managing and updating within the same integrated environment.
   * **Infrastructure as a Service (IAAS):**
   * **Definition:** 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 (e.g., host firewalls).
   * The potential provided to the buyer is to provision processing, storage, networks, and other fundamental computing resources where the buyer is in a position to deploy and run arbitrary software, which may include operating systems and applications. The buyer doesn't 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 (e.g., host firewalls).
   * **Advantages:**
   * **Reduces capital expenditures and optimises costs:** IaaS eliminates the cost of configuring and managing a physical datacentre, which makes it a cost-effective choice for migrating to the cloud. The pay-as-you-go subscription models used by IaaS providers help you reduce hardware costs and maintenance and enable your IT team to focus on core business.
   * **Increases scale and performance of IT workloads:** IaaS lets you scale globally and accommodate spikes in resource demand. That way, you can deliver IT resources to employees from anywhere in the world faster and enhance application performance.
   * **Increases stability, reliability and supportability:** With IaaS, there is no need to maintain and upgrade software and hardware or troubleshoot equipment problems. With the appropriate agreement in place, the service provider assures that your infrastructure is reliable and meets service-level agreements (SLAs).
   * **Improves business continuity and disaster recovery:** Achieving high availability, business continuity and disaster recovery is expensive because it requires a significant amount of technology and staff. But with the right SLA in place, IaaS helps to reduce this cost. It also helps you access applications and data as usual during a disaster or outage.
   * **Enhances security:** With the appropriate service agreement, a cloud service provider can offer better security for your applications and data than the security you would attain in house.
   * **Helps you innovate and get new apps to users faster:** With IaaS, once you have decided to launch a new product or initiative, the necessary computing infrastructure can be ready in minutes or hours, rather than in days or weeks. And because you don't need to set up the underlying infrastructure, IaaS lets you deliver your apps to users faster.
3. **What is CI/CD?**
   * Implementation of Continuous Deployment / Continuous Integration
   * Continuous Integration (CI) is a development practice that requires developers to integrate code into a shared repository multiple times every day. Every check-in is verified by an automated build, which lets developers or teams spot issues early. The benefit of using continuous integration is that it helps identify errors earlier and enables them to be fixed quickly. Keep in mind that one major source of recurring errors is in the code integration step. Use APM and Instrument Your Application you must make sure that an Application Performance Management.
4. **What is APM tool?**
   * (APM) tool is available for the developer and that the application is instrumented to detect issues and performance anomalies. APM tools help in ensuring the smooth operation of your application and ease the process of troubleshooting. Moreover, you can use these tools for custom business telemetry. Be sure to leverage all the rich information that APM can provide about your application.
5. **What is MTTR and MTBF?**
   * Mean Time To Recovery (MTTR) and Mean Time Between Failures (MTBF) are the two metrics that are used to measure system availability:
     1. MTTR represents the average time it takes to reinstate a component after a failure.
     2. MTBF represents the expected time a component can last between outages.
6. **What is KPI’s?**
   * KPI’s are used to increase reliability.
   * These KPIs) can be triggered automation when a threshold is breached. These KPIs should be a measure of business value, not of the technical aspects of the operation of the service. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it’s possible to anticipate and remediate failures before they occur.
7. **Explain N/Tier and Microservices architecture style (Definition, Challenges, Adv and Dis)**
   * **N-Tier:**
   * N-tier is a traditional architecture for enterprise applications. Dependencies are managed by dividing the application into layers that perform logical functions, such as presentation, business logic, and data access. A layer can only call into layers that sit below it. However, this horizontal layering can be a liability. It can be hard to introduce changes in one part of the application without touching the rest of the application. That makes frequent updates a challenge, limiting how quickly new features can be added.
   * N-tier is a natural fit for migrating existing applications that already use a layered architecture. For that reason, N-tier is most often seen in infrastructure as a service (IaaS) solution, or application that use a mix of IaaS and managed services.
   * **Advantages:**
   * It offers portability between the cloud and on-premises as well as between cloud platforms.
   * It has less of a learning curve for most cloud architectures and developers.
   * It has a natural progression from the traditional application model.
   * It’s open to heterogeneous environments, such as Windows and Linux.
   * **Challenges:**
   * You may end up with a middle tier that can only perform CRUD operations on the database, and this can add additional latency with no beneficial work.
   * Its monolithic design stops the independent deployment of features.
   * When you manage IaaS applications, it will demand more work than an application utilizing only managed services.
   * If it’s a large system, the management of network security becomes a hassle.
   * **Microservices:**
   * If your application has a more complex domain, consider moving to a Micro services architecture. A micro services application is composed of many small, independent services. Each service implements a single business capability. Services are loosely coupled, communicating through API contracts.
   * Each service can be built by a small, focused development team. Individual services can be deployed without a lot of coordination between teams, which encourages frequent updates. A microservice architecture is more complex to build and manage than either N-tier or web-queue-worker. It requires a mature development and DevOps culture. But done right, this style can lead to higher release velocity, faster innovation, and a more resilient architecture.
   * **Advantages:**
   * **Agility:** Since microservices are deployed independently, it is easier to handle new feature releases and bug fixes without redeploying the entire application. Also, you can roll back an update if anything wrong or unwanted occurs. In the majority of traditional applications, if a bug is reported in one part, it can block the entire release process. Consequently, you are forced to wait until the bugs are fixed before adding new features.
   * **Small Codebase:** In a monolithic application, there’s an inclination for code dependencies to become twisted over time, since adding a new feature means touching code in several places. As microservices don’t share code or datastores, this style reduces dependencies and makes it easy to add new features
   * **Possibility to Mix Technologies:** Using microservices allows for using multiple technologies and stacks. Each microservice can be built using a stack or a programming language that best suits the team. This does not prevent different microservices from coexisting and interoperating.
   * **Scalability:** Microservices orchestrators, such as Kubernetes, allow for the automatic scaling of microservices, depending on the load. The use of microservices allows for a granularity of the management of the scalability. Only services that need to be scaled are scaled, which optimizes resource usage.
   * **Data Isolation:** It’s a lot simpler to do schema updates, as only a single microservice gets affected. In a monolithic application, schema updates can be challenging because various parts of the application can touch the same data and make risky changes to the schema.
   * **Challenges:**
   * **Complexity:** You will see that microservices applications have more moving parts than monolithic applications. All services are simpler but dealing with the whole system turns out to be a complex task.
   * **Development & Testing:** When you create a small service that depends on other dependent services, you’ll need a different approach than what you would employ to develop a conventional monolithic or layered application. Existing tools aren’t always designed to function with service dependencies; thus, refactoring across service borders can be hard. It’s also a challenge to test service dependencies, specifically when the application is in the process of developing rapidly.
   * **Lack of Governance:** While the decentralized approach to developing microservices has certain benefits, it can result in potential issues. You may end up with several languages and frameworks, and it becomes more difficult to maintain the application. It could be beneficial to put a few project-wide values in place without excessively limiting flexibility in your team’s performance.
   * **Network Congestion & Latency:** The use of numerous small, granular services may result in more interservice communication. If the chain of service dependencies gets very lengthy, i.e., service A calls B, which calls C, and so on, the extra latency can be an issue. Being a developer, you must design APIs with care. Prevent excessively chatty APIs, consider using serialization formats, and find places to employ asynchronous communication patterns.
   * **Data Integrity:** Every microservice is responsible for its own data tenacity. Consequently, you may face issues in maintaining data consistency. Therefore, it is recommended that you embrace ultimate consistency where it’s conceivable for you.
   * **Management:** It is obvious that you would like to make your microservices project a success, and you can do this with a mature DevOps culture.When you try correlated logging across services, it will be a very challenging task. Characteristically, to handle this situation, you must correlate logging with manifold service calls for single-user operation.
   * **Versioning:** As a developer, you must realize the importance of updating a service without disturbing services that rely on it. You can update various services at any time, and if you do so without careful design, certain issues with backward or forward compatibility can occur.
   * **Skillset:** Microservices help you create exceedingly distributed systems. Therefore, you must ensure that you have a highly skilled and experienced team of developers to complete your project effectively and successfully.
8. **What is Server? Explain different types of Servers?**
   * **Definition:** A server is a piece of computer hardware or software that provides functionality for other programs and devices. Server provides various functionalities, often called services such as sharing data or resources among multiple clients, or performing computation for client. Single server can serve multiple clients, and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device. Servers are available in various types.
   * **Types of Servers:**
   * **Application servers:** Application server is a server which is used to host the application. Application server are system software upon which web application or desktop applications run. It consists of web server connector, computer programming language, database connector, runtime libraries and the Administration code. Application server frameworks are Software framework for building application server. It allows users to create both web application and server environment.
   * **Client servers:** Client server is a distributed application that partition task or workload between the providers of resource or service called servers and service requesters are called as client Both clients and server communicate with each other over computer network on separate software hardware, but both client and server by reside in the system. Client does not share any of its resources but request resources from server
   * **Collaboration Servers:** The goal of collaboration server is to support conduct of shared activities of software development teams. Its main concern is to support project management activities rather than project products. Collaboration server provide scheduling support for each project, team or individual. Also provides alerts for schedule reviews and collaboration event.
   * **Mail Servers:** Mail server is server that handles and delivers email over a network, usually over the internet. It receives email from client computers and deliver them to other mail server. It also
   * delivers email to client computers. Client computer is normal computer where user read their emails for examples computer at home or office. Also advanced mobile or smartphones with email facilities are considered client computers.
   * **FTP Servers:** FTP server is used for transferring files supporting sub directories, log in and set of manipulation commands. It can also be used to upload HTML pages on HTTP server or download log files to remote PC. Access control interface allows users to start or stop FTP server and to manage the built-in user accounts. It uses user callback function to send the notification about FTP server to user application.
   * **Proxy Servers:** Proxy server is a server application that acts as an intermediary for request from clients seeking resources from servers that provide those resources. Proxy server functions behalf of the client when requesting service, potentially masking the true origin of the request to the resource server. Proxy server are used as content control software, to filter the encrypted data, logging, and eavesdropping, repairing errors, accessing services, cross-domain resources, security, etc.
   * **Telnet Servers:** Telnet server is an application protocol used to provide bidirectional interactive text orientated communication service using virtual terminal connection over internet. Telnet client application are available for virtually all computer platforms. On remote host, telnet server provides access to the command-line interface. Using TCP, user data and telnet control information is transferred on data connection.
   * **Real Time Communication Server:** Real time communication server is also known as chat server or IRC server. It refers to instant messaging servers. Real time communication server allows large number of users to exchange the information near instantaneously. For example, call over mobile phone or landline, instant messaging like WhatsApp, Facebook, Instagram, Live tv news, video calls, etc.
   * **Open Source Servers:** Open source server is a public domain software designed to deliver web pages over www. It runs on computers that is connected to the internet. Open source servers support file uploading and downloading using FTP, provides security features. It also supports other communication protocols.
   * **Web Server:** Web server is a computer software and underlying hardware that Accept request through HTTP. HTTP is a network protocol which is used to distribute the web pages. Using HTTP web browser send request to the server and initiate the communication. Then web server responds with the resources or content or error message. Web server also accepts and store the resources sent from the user agent.
   * **Virtual Servers:** Virtual server is a server that is located in an offsite data center and its resources are shared by multiple users who have control over it. In other words, we can say that virtual server converts one physical server into multiple virtual machines where each can run their operating systems. Virtual servers are a more efficient use of power and can increase a server’s utilization from the typical 15 percent to as much as 80 percent. They are used for variety of applications from remote file access to web design and development.
9. **What is CapEx&OpEx. Explain importance of CapEx&OpEx cloud?**
   * **CapEx Definition:**
   * CapEx is defined as business expenses to gain long-term benefits by buying assets such as buildings or equipment. CapEx examples of IT infrastructure are an office building, data center, physical servers, desktops, laptops, storage, printers, scanners, generators, software, and other related assets. In the CapEx type of business expense, you invest once, and you can reap benefits of that business expenses many years in the future.
   * **OpEx Definition:**
   * Operational expenses are expenses to run day-to-day business operations like services and consumable items that get used up, and you need it again to run the business. Examples are stationery supplies, printer cartridges, utility bills, monthly office rent, domain name registration and renewal, website maintenance, etc. The main idea is that these things are needed to run the businesses, but they are not considered long-term investments like items in CapEx.
   * From the business accounting perspective, operational expenses can be deducted from the revenue, thus decreasing your tax liability and increasing your profit. Another Another advantage of OpEx is that if it is not working as intended, you can stop using it or replace it with another alternative, which is tricky in CapEx. On the other hand, one potential drawback of OpEx is that sometimes it is challenging to gauge value proposition because of the short-term nature of OpEx – which is typically a year from the business accounting perspective.
   * **Importance of CapEx and OpEx in cloud:**
   * In the above discussion for OpEx and CapEx, it's clear there are steep differences in OpEx and CapEx both from the accounting perspective and gauge of value proposition provided in both of these financial models of IT expenditure. The question is what their use cases are. In other words, when to use one over the other.
   * If you are tight on capital expenditure or want to start very quickly as what you are looking for is already available on the public cloud, the public cloud's pay-as-you-go model could be the right choice –that would be the OpEx financial model. On the other hand, if you would like to have complete control of IT resources, you would instead prefer to build a private cloud -- this would be the CapEx financial model. In this private cloud case, your organization will be responsible for all expenditures.
   * The current trend is using the public cloud as this is where there are actual multi-facet advantages of the cloud. However, organizations with tight, unique security and regulatory requirements tend to think critically about public cloud adoption.
   * In some cases, these organizations adopt a middle ground of hybrid cloud where they migrate those applications which are non-essential and not in the tight security and regulatory requirements realm. But keep critical and essential applications on private cloud or on-premises data center as it is.
10. **What is Server Failover? Difference between Active-Standby and Active-Active Failover?**
    * **Definition:** Server failover is the practice of having a backup server (or servers) prepared to automatically take over if the primary server goes offline. Server failover works like a backup generator. When the power goes out in a building or home, a backup generator temporarily restores electricity. Similarly, in server failover, a secondary server takes over when the primary server fails. The goal of server failover is to improve a network or website's fault tolerance, or its ability to continue operating when one of its parts fails.
    * A server's primary job is to store content and data to share with other computers. While there are different types of servers, web servers are perhaps the most well-known because they keep websites and applications operational. When web servers fail, they cannot process requests, which means they cannot serve data to clients. Without server failover, a failed server can cause a loading error or a site outage.
    * **Active-standby Failover:**
    * In active-standby, there is a primary server and one or more secondary servers. In a two-server setup, the secondary server monitors the primary one but otherwise remains inactive. If the secondary server senses any change with the primary server, it will take over and advise the data center that the primary server needs restoration. Once the primary server is restored, it takes over once again, and the secondary server resumes a standby position. The act of a primary server resuming operations is called failback.
    * **Active-active Failover:**
    * By contrast, in a two-server active-active configuration, both servers must remain active. An active-active configuration is typically associated with load balancing because the servers are configured in the same way and share the workload. When a server fails in an active-active configuration, the traffic routes to the operational server or servers.
11. **Explain Uptime Data Center and their Tier I/Tier II/Tier III/Tier IV classifications?**

* **Uptime Data Center:** Uptime Institute created the data center Tier classification levels over 25 years ago, and today, they remain the international standard for data center performance. Our data center Tier definitions explain the infrastructure required for data center operations. There are different Tiers according to the system availability needed. These classifications are objective and reliable methods for comparing the performance of one site infrastructure against another and aligning infrastructure investments to business goals.
  + **Tier I**
  + A Tier I data center is the basic capacity level with infrastructure to support information technology for an office setting and beyond. The requirements for a Tier I facility include:
    1. An uninterruptible power supply (UPS) for power sags, outages, and spikes.
    2. An area for IT systems.
    3. Dedicated cooling equipment that runs outside office hours.
    4. An engine generator for power outages.
  + Tier I protects against disruptions from human error, but not unexpected failure or outage. Redundant equipment includes chillers, pumps, UPS modules, and engine generators. The facility will have to shut down completely for preventive maintenance and repairs, and failure to do so increases the risk of unplanned disruptions and severe consequences from system failure.
  + **Tier II**
  + Tier II facilities cover redundant capacity components for power and cooling that provide better maintenance opportunities and safety against disruptions. These components include:
  + 1. Engine generators.
  + 2. Energy storage.
  + 3. Chillers.
  + 4. Cooling units.
  + 5. UPS modules.
  + 6. Pumps.
  + 7. Heat rejection equipment.
  + 8. Fuel tanks.
  + 9. Fuel cells.
  + The distribution path of Tier II serves a critical environment, and the components can be removed without shutting it down. Like a Tier I facility, unexpected shutdown of a Tier II data center will affect the system.
  + **Tier III**
  + A Tier III data center is concurrently maintainable with redundant components as a key differentiator, with redundant distribution paths to serve the critical environment. Unlike Tier I and Tier II, these facilities require no shutdowns when equipment needs maintenance or replacement. The components of Tier III are added to Tier II components so that any part can be shut down without impacting IT operation.
  + **Tier IV**
  + A Tier IV data center has several independent and physically isolated systems that act as redundant capacity components and distribution paths. The separation is necessary to prevent an event from compromising both systems. The environment will not be affected by a disruption from planned and unplanned events. However, if the redundant components or distribution paths are shut down for maintenance, the environment may experience a higher risk of disruption if a failure occurs.
  + Tier IV facilities add fault tolerance to the Tier III topology. When a piece of equipment fails, or there is an interruption in the distribution path, IT operations will not be affected. All of the IT equipment must have a fault-tolerant power design to be compatible. Tier IV data centers also require continuous cooling to make the environment stable.

1. **What is Data Center and Explain its different types. (Googled Answers)**
   * **Definition:** A data center -- also known as a datacenter or data centre -- is a facility composed of networked computers, storage systems and computing infrastructure that organizations use to assemble, process, store and disseminate large amounts of data. A business typically relies heavily on the applications, services and data contained within a data center, making it a critical asset for everyday operations.
   * Enterprise data centers. These proprietary data centers are built and owned by organizations for their internal end users. They support the IT operations and critical applications of a single organization and can be located both on-site and off-site.
   * Managed services data centers. Managed by third parties, these data centers provide all aspects of data storage and computing services. Companies lease, instead of buy, the infrastructure and services.
   * Cloud-based data centers. These off-site distributed data centers are managed by third-party or public cloud providers, such as Amazon Web Services, Microsoft Azure or Google Cloud. Based on an infrastructure-as-a-service model, the leased infrastructure enables customers to provision a virtual data center within minutes.
   * Colocation data centers. These rental spaces inside colocation facilities are owned by third parties. The renting organization provides the hardware, and the data center provides and manages the infrastructure, including physical space, bandwidth, cooling and security systems. Colocation is appealing to organizations that want to avoid the large capital expenditures associated with building and maintaining their own data centers.
   * Edge data centers. These are smaller facilities that solve the latency problem by being geographically closer to the edge of the network and data sources.
   * Hyperscale data centers. Synonymous with large-scale providers, such as Amazon, Meta and Google, these hyperscale computing infrastructures maximize hardware density, while minimizing the cost of cooling and administrative overhead.
2. **What is VLAN and explain its types.**
   * **VLAN Definition:**
   * A VLAN is a virtual network formed through the logical grouping of devices on a LAN. The devices are, typically, switches, and the LAN is Ethernet. With LAN, a network packet is received by all devices on it. With VLAN, the network packet is sent to only a specific set of devices that constitute what is called, broadcast domain. VLANs partition the network at Layer 2.
   * LAN uses protocols such as Token ring and FDDI (Fiber Distributed Data Interface). VLAN uses IEEE802.1q and Inter-Switch Link (ISL) protocols [2]. The Ethernet LAN represents the collision domain on which Layer 0 Ethernet (CSMA/CD) packets collide. A VLAN represents the broadcast domain, i.e., a group of devices configured to receive broadcast traffic (Layer 1 Data Link Frames) from one another. Without VLANs, a broadcast message sent from a host reaches all network devices increasing CPU overhead on each device and reducing the overall network security. With VLAN configuration, a broadcast from the host is limited to devices on the VLAN.
   * A VLAN can be created from one or multiple LANs. It enables the network administrator to automatically limit access to a specified group of servers/desktops into different isolated network segments.
   * **Two types of VLANs**
   * 1. Port-based (Untagged) VLANs: A single physical switch is simply split into multiple logical switches.
   * 2. Tagged VLANs: Multiple VLANs use a single switch port. Tags are attached to the individual Ethernet frames as they exit the port. Tags contain the VLAN identifiers specifying the VLAN to which the frame belongs. When both switches understand tagged VLANs, the connection can be accomplished using a single cable connecting from, what is called, a “trunk” port.
3. **What is Subnetwork?**
   * Subnetwork or subnet is a logical network partition at Layer 3 of the OSI model. At Layer 3, i.e., at the Network layer, each computer or host has at least one IP address as a unique identifier. The Internet Protocol (IP) is used for sending data from one computer to another over the internet. A subnet is defined to subdivide large IP networks into smaller, more efficient subnetworks. A subnet aims to divide a large network into smaller, interconnected networks to minimize the broadcast traffic on a single network segment, thereby improving available network bandwidth. It also optimizes the usage of available IP address space.
   * Subnet masks split an IP address into bits that identify the network and host parts. When a device sees the network identification and host identification bits of another device’s IP address, it can determine if it is part of the same network or some other network. Figure 6.5 shows the structure of the IP address and the network identification bits and host identification bits.
4. **What is Subnetting?**
   * Subnetting is the segmentation of a network address space. It allows its connected devices to communicate with each other. Routers are used to communicate between subnets. Subnet masks specify the range of IP addresses used within a subnet. Two types of subnet specifications have been in existence:
   * 1. Classful: Three classes of subnets have been defined as summarized in Table 6-2. Older and less used. Classes disproportionately distribute the number of available IP addresses.
   * 2. Classless: Classless Inter-Domain Routing (CIDR) notation is used to specify a subnet, as summarized in Table 6-3. A trailing “/” slash and a number are used that specify how many bits are used to identify the network portion of the address. Currently widely used.
5. **What is North-South Traffic vs East-West Traffic?**
   * **North-South traffic:** For traffic from a server in a data center to reach the internet, it needs to reach WAN. The traffic that involves the inward and outward flow of data packets from the server (LAN) to WAN is north-south traffic. Traditionally, most of the traffic in the past in data centers was of this kind.
   * **East-West traffic:** The traffic flow within a data center, VLAN, or subnet is referred to as east-west traffic. For example, the data communication from the application server to the DB server constitutes east-west traffic. This kind of traffic is the norm in contemporary data centers.
6. **What is Mean Time Between Failures (MTBF)?**
   * Average time between system failures. MTBF is related to system uptime and not being under the control of the operations team.
7. **What is Mean Time to Repair (MTTR)**
   * Mean Time To Repair (MTTR): Average time to troubleshoot, repair, and restore the system from failure. MTTR is related to downtime a system can tolerate to comply with availability criteria.
   * Availability - (MTBF)/(MTBF+MTTR) or (Uptime/Uptime + Downtime)
8. **Explain Disaster Recovery Process.**
   * Disaster Recovery is reacting to a disaster incident. However, a proactive step before that involves preparing for a disaster scenario, called business continuity planning. Only when business continuity planning is done right can disaster recovery (DR) be successful with the alignment of business planning, architecture strategy, solutions, and operations. Disaster recovery, therefore, has two aspects to it - preparation and execution.
   * **Preparation**
     1. Establish DR operational characteristics: RPO and RTO are two key characteristics.
     2. Setup alternate DR site, which is referred to as secondary site, with the right model:
        1. a. Active-active: Both primary and secondary data center run applications (workloads) in production environment and serve user requests. A load balancer distributes user requests to both the data centers.
        2. b. Active-passive: Primary data center functions as an active application site, serves user requests, and replicates critical business data to secondary. The secondary data center is ready to be activated to service applications should the primary data center fail due to some reason.
     3. Form DR operations team with a clear and well-documented approach to be employed in the event of a disaster. The DR team-
        1. a. Maintains DR infrastructure and automation.
        2. b. Defines procedures and processes to be employed during DR.
        3. c. Conducts DR tests at regular intervals to ensure that organization is DR ready.
        4. 4. Establish procedures to replicate data to the DR site for identified applications:
        5. 5. Declare DR when the disaster event occurs.
   * **Execution**
   * All the preparations for DR will come to fruition should a DR event occur. The activities to be carried as part of DR execution are:
   * Recover shared infrastructure platforms and services.
     1. a. Shared infrastructure platforms: Network components, servers, block storage (SAM), file storage (NAS), and object storage.
     2. b. Shared infrastructure services: DNA, AD, Citrix, and so on.
   * 2. Recovery shared application platforms and services.
     1. a. Shared application platforms: Virtualization servers (ESXi hosts) or private cloud platforms, Mainframe, Mid-range (AIX and IBMi), SQL Server, Oracle RDBMS, IBM DB2, and so on.
     2. b. Shared application services: Tomcat, WebShpere, MQ-Series.
   * 3. Recover business application and services.
     1. a. Provision infrastructure for business applications to be brought up in the secondary site.
     2. b. Deploy applications and related services.
     3. c. Attach storage with the correct data to the compute.
   * 4. Test and switch all traffic to the secondary site.
   * **Replication for DR**
   * Backup is making copies of data and preserving it at one more locations for operational recovery, i.e., should the actual data be corrupted or unavailable for some reason. On the other hand, replication is copying the data from the primary data center to the secondary data center for disaster recovery.
     1. **Host (or Guest-OS) based replication:** Replication software is deployed on a physical or virtual server, and the data is replicated using the software (e.g. Vision Solutions DoubleTake, Veritas Volume Replicator).
     2. **Appliance-based replication:** Appliances are deployed in primary and secondary sites. Data that needs to be replicated is duplicated in the appliance and replicated (e.g., IBM Spectrum Protect server, DELL/EMC Recover Point)
     3. **VM-Snapshot replication:** VM-level snapshots are taken at the primary site and sent to the secondary site. They are point-in-time snapshots created in the hypervisor and replicated asynchronously.
     4. **Hypervisor-based replication:** The replication software plugs into the hypervisor and copies VM-level data from the primary site to the secondary site. The data in the secondary site is continuously kept up to date either synchronously or asynchronously with changes made on the primary site (e.g. Zerto Enterprise Cloud Edition for VMs, VMware SRM).
     5. **Array-based (or storage-based) Replication:** The data in SAN storage array (e.g. DELL/EMC PowerMax) and NAS/iSCSI storage arrays may be replicated at storage-level by employing specific tools for replication.
9. **Explain three different types of backups?**
   * **Full back-up:** Backup of all files, objects, bytes. It represents a complete backup of all data and can be used for recovery without additional efforts. It takes time to do a full backup, depending on the size of the data and the number of systems on which full backup needs to be done.
   * **Differential backup:** Differential backup makes a copy of data that has changed since the full backup. During restore, the last full backup is used, and then the differential backup is applied on top of it, thus saving time. However, as the number of days from full backup increases, the data to be backed up also increases, resulting in an increase in time taken to take differential backup.
   * **Incremental backup:** Backup is taken of changes made since the last backup (full or incremental). The last full backup is used during restoration, and subsequent incremental backups are applied in the correct order. This process takes the least amount of backup window available in most enterprises.
10. **Explain Block Storage/File Storage/Object Storage?**
    * There are three types of storage solutions in use, both on-premises and cloud environments.
    * 1. Block Storage
    * 2. File Storage
    * 3. Object Storage
      1. **Block Storage**
    * Block storage is a type of storage in which data is stored in fixed-size blocks. Metadata is maintained by the storage system for each block. A software program for the storage system uses metadata to locate the desired blocks for data retrieval. A raw volume of data storage from the constituent blocks is made available to a physical server (with additional storage controllers) or virtual machine. Each raw volume can function as an individual hard drive or storage repository. It can be mounted as a volume or mapped as a drive for access from the operating system.
      1. **File Storage:**
    * Many files are generated both by individuals and applications in any organization. The file storage solution is an effective mechanism to store and retrieve file-based data from a system that provides access to operating systems as a mount point or drive mapping.
      1. **Object Storage:**
    * When custom metadata is stored along with the actual data to facilitate retrieval, it represents an object. Mechanisms to store and retrieve such objects are part of object storage solution. Objects are typically unstructured without hierarchies, in a unified addressed space, and accessed using REST APIs over HTTP. With the increasing need for enterprises to deal with unstructured data (images, audio, and video) and the increasing demand for storing, retrieving, and transmitting them, object storage has become critical for many businesses. This form of storage is best suited for data that does not change and is widely offered as cloud storage by cloud providers.
11. **What is an Intrusion Prevention System/Intrusion Detection System?**
    * An Intrusion Detection system (IDS) is a system that monitors network packets and raises alerts on detecting known signatures of threats such as malware.
    * Intrusion Prevention System (IPS) takes alerts to the next level of controls by rejecting network packets representing security threats. Above figure depicts the deployment of IDS/IPS in the DMZ before the traffic reaches the web server.
12. **What is Web Application Firewall?**
    * A web application firewall (WAF) is an application firewall for HTTP applications. It is like a reverse proxy as it protects servers that host one or more web applications by inspecting and filtering traffic between each web application and the internet. In above figure WAF is shown deployed in the DMZ as an appliance that examines HTTP traffic before it reaches the web server. It may also be deployed as a server-side software plugin or packaged as a cloud service to detect and filter threats that could expose online applications to denial-of-service (DoS) or degrade performance. WAFs may be stateless or stateful.
    * Examples of attacks that WAF may be used to filter are SQL injection, Cross-site Scripting (XSS), Layer 7 DoS, Cookie poisoning, Web scraping, and Unvalidated input
13. **What is Data Leakage Prevention (DLP)?**
    * Data Loss Prevention (DLP) involves the detection and prevention of data loss through unwanted destruction of sensitive data (e.g., ransomware attack) and data leakage by the unauthorized transmission of data from within an organization to an external recipient. DLP threats occur through the web, email, data storage devices. DLP solutions enable discovery and control of sensitive data by setting business rules that classify confidential information to prevent disclosure maliciously or accidentally.
14. **What is SAST/DAST/IAST?**
    * **SAST**
      1. Static application security testing is a white box testing method to test code for vulnerabilities.
      2. Typically, it plug’s into IDEs to run scans on code.
      3. Examples of vulnerabilities that can be identified through SAST are:
         + 1. SQL injection
           2. Cross-site scripting
           3. Buffer Overflows.
      4. Examples of SAST tools are klocwork, SpectralOps, Veracode
    * **DAST**
      1. Dynamic application security testing (DAST) is a black-box testing method performed by running tests on on application in execution.
      2. Simulates external attacks and identifies vulnerabilities based on outcomes.
      3. Examples of vulnerabilities that can be identified through DAST are:
         + 1. Path traversal
           2. Insecure server configuration
           3. SQL/Command injection
           4. Cross-site scripting
      4. Examples of DAST tools are Ac0unetix, AppScan, Netsparker
    * **IAST**
      1. Interactive Application Security Testing (IAST) is a white box testing method on an application instrumented with specific interfaces to identify vulnerabilities.
      2. IAST testing is performed in real-time while the application is running to identify the problematic code lines from a security perspective and notify the developer for remediation
      3. All the OWASP vulnerabilities may be identified with IAST tools
      4. Examples of IAST tools are: Hdiv Detection (IAST), Synposis Seeker, Veracode
15. **Explain any 5 Cryptographic Attacks?**
    * **BFA (Brute Force Attack) –** Here, the intruder tries to find the decryption key by entering all possible information. For instance, the key contains 8 bits. That means the total number of possible keys is 256 (i.e. 28). The attacker tries all of these keys in order to obtain the plaintext. The longer the key, the longer the time needed for successful decryption.
    * **COA (Ciphertext Only Attack)** – This tactic requires the complete set of ciphertexts used for a message. When COA gets the plaintext from the given ciphertexts, the tactic is considered successful. Attackers may also get the corresponding encryption key using this attack.
    * **CPA (Chosen Plaintext Attack)** – This attack requires the attacker to work on the plaintext he/she selected for encryption. Simply put, the attacker has the plaintext-ciphertext combination. It means the task of decrypting the information is easy and simple. It is the first part of the attack – convincing the sender to encrypt certain pieces of information – that presents the most difficulties.
    * **KPA (Known Plaintext Attack)** – With this tactic, the attacker should know some parts of the plaintext. He/she has to use this knowledge to obtain the rest of the message.
    * **Birthday Attack** – This is a subtype of the brute force approach. Attackers use this tactic when working against cryptographic hash functions. Once the intruder finds two inputs that produce similar values, a collision is said to occur: the hash function is broken and the system is breached.
    * **MIM Attack (Man in the Middle Attack)** – This attack is particularly designed for public key cryptosystems. In general, these systems require the exchange of keys before the actual transmission of the ciphertext.
    * **SCA (Side Channel Attack)** – This attack is used to exploit the weaknesses of a cryptosystem’s physical implementation. Here, the attackers ignore the system’s algorithms and digital protection.
    * **Fault Analysis Attacks** – When using this attack, the intruder looks for errors produced by the system. He/she uses the resulting information to breach the system’s defenses.
    * **Timing Attacks** – Here, attackers use the fact that different calculations require different processing times. These people can acquire some data about the message processed by a computer system. They do this by measuring the time used by the computer in performing its calculations.
    * **Power Analysis Attacks** – These attacks are similar to the previous one. However, instead of time, they use the amount of power consumed by the computer system. This information is used to determine the nature of the plaintext.
16. **What is IAM Policy?**
    * IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API
17. **What is Phishing/Malware/Social engineering/DDos/APTs/MitM/Ransomware/Password Attacks?**
    * **Phishing:** Fraudulent email or messages meant to deceive users as being reputable or known sources to steam sensitive data.
    * Tools for protection - SpamTital, Proofpoint, Mimecast
    * **Malware:** Malicious software that can be used to cause harm to a user. Viruses, worms, Trojans, and spyware are different forms of malware.
    * Tools for protection - Avast Antivirus, Kaspersky, Trend
    * **Social engineering:** Uses human interaction to trick users into revealing sensitive information.
    * Tools for protection - Policies, Training programs for users.
    * **Distributed denial-of-service (DDoS) attacks:** Flooding the target system with messages from multiple sources to disrupt the traffic of the largest system to prevent it from functioning or affect its performance.
    * Tools for protection - SolarWinds Security Event Manager, AWS Shield, Indusface AppTrana
    * **Advanced persistent threats (APTs):** Sustained targeted attacks to infiltrate a network and remain undetected for an extended period to steal data.
    * Tools for protection - Security Information and Event Management (SIEM) Tools such as SolarWinds Security Event Manager, Splunk Enterprise Security.
    * **Man-in-the-middle (MitM):** Attacks involve an interception and relay of messages between two parties who believe they are communicating.
    * Tools for protection: Hetty, Bettercap, Mitmproxy
    * **Ransomware:** Involves locking the user’s computer system files and demanding a payment to unlock them.
    * Tools for protection: Bitdefender Antivirus, AVG
    * **Password Attacks:** As the password is the most used mechanism to authenticate users to a system, getting to know the right password is a common attack approach. Brute-force attacks and Dictionary attacks are

often used by hackers to get to know the password by trial-and-error approach. Password policies, including account lockout, password change at regular intervals, and password complexity, mitigate password attacks.