**Identify the key aspects for a successful service discovery solution**

All the options mentioned

**Service Discovery can perform the following for the services**

All the options mentioned

**Which of the registration patterns is best suited for complex architecture**

Third party registration pattern

**Identify the right service registry solutions that supports discovery**

Zookeeper

**In \_ pattern, clients connect with the backend services directly**

Client side discovery pattern

**\_ supports load balancing in addition to service discovery**

Ribbon

**Is it possible for any service discovery tool to satisfy all the three guarantees (Consistency, Availability, Partition) in a distributed environment?**

No. Only two of the guarantees can be achieved

**In distributed systems, to service any Read / Write request, a leader node is mandatory. Is this statement correct**

No. Leader node is required for only for write request

**The process of consensus in distributed systems is essential to achieve**

All the options mentioned

**In distributed systems, knowing that partition can always occur, what kind of service discovery tool would you prefer?**

One that follows AP (Availability, Partition)

**In server side discovery pattern, load balancing and routing of the request to the service instances is taken care of by the service discovery tool itself. Is it true or false?**

True

**Service registration can happen by how many ways**

3

**Can an external service or process poll for all the services and update the service registry for discovery?**

Yes, it is possible

**How can one ensure high availability of service discovery solution so that services can be easily discovered even in the event of any failure scenarios. (I) Service registry information exchange among clusters. (II) Service clients caching the service registry information**

I & II

**\_ is the foundation for proper monitoring, load balancing and routing in distributed systems**

Collecting the application functionality details

**\_ is associated to every user request to track the service requests placed by the users**

Instance Id

**Identify the correct properties of consensus**

All the options mentioned

**AWS Elastic Load Balancer(ELB) is an example following \_ pattern of service discovery**

Server side discovery pattern

**\_ is an election algorithm in which a coordinator is elected by highest process id silencing the other nodes in the system**

Palos Algorithm

**What happens when heart beat from any service instance is not received by the service register**

The instance is disregarded for service availability