Q.2 SERVICE LOAD BALANCING

→ It is considered a specialized variation of workload distribution for scaling cloud services specifically for scaling service implementation.

→ Redundant deployments of cloud services are created, with a local balancing system added to dynamically distributed workloads.

→ The duplicate cloud service implementation are organized into a resource pool while load balancer is positioned as either an external or buit in component to allow the host servors to balance the workload themselves

→ Depending upon the anticipated workload and processing capacity of host server environment, multiple instances of each cloud service implementation can be generated as part of a resource pool that responds to thictating request volumes more efficiently

→ The load balancers can be positioned either independent of the cloud services and their host servies or buit in as part of the application or servers environment in the lower case a primary server with load balancing logic can communicate or servers's environment in workload.

→ The service load balancing ~~logic~~ architecture can involve the following mechanism:

1. CLOUD USAGE MONITOR - cloud usage monitors may be involved with monitoring cloud services instances and their respecti. it resources consumption levels.

2. RESOURCE CLUSTER - active cluster groups are incorporated in this architecture to help balance workload across different members of cluster.

3. RESOURCE REPLICATION - It is utilized to generate cloud service implementations in support of load balancing requirements.

Q3 DOCKER ARCHITECTURE - It follows client-server architecture which includes the three main components that are Docker Client, Docker Host and Docker Registery

1. DOCKER CLIENT - It uses command and rest API to communicate with docker daemon. When a client runs any docker command on docker client terminal, the client terminal send these docker commands to docker daemon. Docker daemon receives these commands from docker client in form of REST API's request.

2. DOCKER HOST - It is used to provide an environment to execute and run applications. It contains docker daemon, images, containers, networks and storage.

3. DOCKER REGISTRY - Docker registry manages and stores images.

There are 2 types of registeries in docker:

- PUBLIC REGISTRY - It is also called as docker hub.
- PRIVATE REGISTRY - It is used to share images within the enterprise

other docker objects are - DOCKER IMAGES
- DOCKER CONTAINERS
- DOCKER NETWORKING
- DOCKER STORAGE

Docker differs from virtual machies in following ways:

→ OS SUPPORT & ARCHITECTURE

- VMs have host os and the guest os inside each VM. Guest os can be any os like Linux or Window.
- Docker containes host on a single physical server with host os, which are shared among them

→ SECURITY

- VMs are stand alone with their kernel and security features
- On the flipside, providing root access to application and running thew with administrative premisis is not reccommended in docker container

→ PERFORMANCE

- VMs are more resource intensive than docker
- Lightweight architecture of docker container is less resource intensive than VMs