**Swarm Microservice Deployment**

**Description**

**Objective**

To deploy a scalable, multi-service voting application on a manager node, ensuring efficient orchestration, fault tolerance, and seamless monitoring through Docker visualizer

**Solution:**

For deploying a scalable, multi-service voting application using Docker Swarm and integrating DockerVisualizer for monitoring. The stack will include multiple services such as Frontend, Backend (API), Database, and Worker services. The docker-compose.yml file defines the services and integrates the Docker Visualizer for real-time insights.

**Tools to be used:**

The following tools used in this project serve specific purposes within the industry:

1. Docker swarm: The stack is deployed using Docker swarm, a container orchestration tool that allows you to manage a cluster of Docker nodes and deploy services across them.

2. Docker microservices: These are small and independent services that run in separate containers, each handling a specific function within an application. This architecture allows for modular development, scalability, and easier maintenance.

3. Swarm cluster: It is a group of Docker nodes working together as a single system to deploy and manage services. It provides built-in orchestration, ensuring high availability, scalability, and efficient load balancing across containers.

4. Docker compose: It is used to define and manage multi-container Docker applications. It specifies the services, networks, and volumes required for the application.

Steps Include:

1. Initialize Docker Swarm on the manager node.

Sudo docker swarm init

1. Provide the input in the two worker nodes.
2. Create the docker-compose.yml File with the following content:
3. Run the following command to deploy the stack on the Docker Swarm manager node:

Sudo docker stack deploy --compose-file docker-compose.yml multi-tier-app

1. Check the status

Sudo docker stack ps multi-tier-app

1. View running services:

docker service ls

1. You can dynamically scale services using:

Sudo docker service scale multi-tier-app\_frontend=4

Sudo docker service scale multi-tier-app\_backend=3