QUESTION:-

Is docker-compose file always written for a common network ? Can two containers already running in different networks be brought to talk to each other using docker-compose ?? If the answer is Yes, Any real-time scenario to support the argument?

ANSWER:-

No, a Docker Compose file is not always written for a common network. Docker Compose allows you to define multiple networks within the same `docker-compose.yml` file, and containers within those networks can communicate with each other.
Yes, two containers running in different networks can be brought to talk to each other using Docker Compose. Docker Compose allows you to specify which containers belong to which networks, and you can configure communication between containers across different networks using their service names.

Here's a real-time scenario to support the argument:

Let's consider a microservices architecture where different services are deployed as separate containers, each residing in its own Docker network for isolation purposes. However, these services need to communicate with each other. Docker Compose can be used to define these services and their networks.

```
version: '3'
services:
  web:
    image: nginx
    networks:
      - frontend

  api:
    image: my-api-service
    networks:
      - backend
      - frontend

networks:
  frontend:
  backend:
```

In this example, we have two services: `web` and `api`. The `web` service is in the `frontend` network, while the `api` service is in both `backend` and `frontend` networks. This configuration allows the `api` service to communicate with both the `web` service (in the `frontend` network) and other backend services (in the `backend` network).
By default, Docker Compose creates a bridge network for each service and also allows you to define custom networks. This flexibility enables containers from different networks to communicate with each other within the same Docker Compose setup.