1. mkdir app
2. cd app
3. code .
4. package.json

```
{
    "dependencies": {
        "express": "*",
        "redis": "2.8.0"
    },

    "scripts": {
        "start": "node index.js"
    }
}
```

5. index.js

```
const express = require('express');
const redis = require('redis');

const app = express();
const client = redis.createClient();
client.set('visits', 0);

app.get('/', (req, res) => {
    client.get('visits', (err, visits) => {
        res.send('Number of visits is  = '  + visits);
        client.set('visits', parseInt(visits) + 1);
    });
});
```

```
app.listen(8081, () => {
console.log("listening on port 8081");
});
```

6. Dockerfile

```
FROM node:alpine
WORKDIR '/app'
COPY package.json  .
RUN npm install
COPY . .
CMD ["npm", "start"]
```

7. docker build .
8. docker build –t priya123456/nodeapp:latest
9. Try to run container from the built image : docker container run priya123456/nodeapp
   You observe that container starts with an error

10. Run the redis container : docker container run redis

Introducing Docker Compose

1. docker-compose.yml file

```
version: '3'
 services:
   redis-server:
      image: 'redis'
   node-app:
       build: .
       ports:
         -   "4001:8081"
```

Networking in docker compose file

2. Now when the node-app should start, it will look for redis server. As both containers will be started by docker-compose, the name of the redis service in docker-compose.yml file will identify the redis-server running in the container.

Modify the index.js file with redis connection string.

```
const client = redis.createClient({
    host: 'redis-server',
    port: 6379
});
```

Docker Compose commands

1. Command : ls
   Output: Dockerfile index.js package.json docker-compose.yml

2. Docker-compose up

3. Verify both container details: docker ps

4. Access web page: http://<<ip address of linux machine>>:4001

   Number of visists = 1

   Keep refreshing the page

5. To start containers in the detach mode: docker-compose up –d

6. To view containers: docker ps OR docker-compose ps

7. To stop all containers: docker-compose down