

Solution

The solution to exercise 4

We'll cover the following



- Steps for basic troubleshooting
- Issues in services
 - Web
 - Database
 - Visualizer

Troubleshooting is really fun because you learn more while fixing issues and you may end up fixing extra issues generated by you while fixing an existing one.

Let's walk through the solution and check whether you fixed exact issues or ended fixing new ones also.

Steps for basic troubleshooting

- Check for network connectivity between services
- Check for port mappings
- Check for any required volume mounts
- Check for valid image name to start services
- Check if the network used for connectivity exists
- Only supported are commands
- Check for any application-specific issues such env variables or init scripts

Issues in services

We will go through one service at a time.

Web

In web services,

- The very first thing we can see is the **links** keyword. In swarm mode, instead

of `links`, we should use `networks` to establish communication between services. So, remove `links` and write

```
networks:  
  - app
```

to connect the database to the app.

- What we need to do next is not an issue if you can increase replicas using `deploy` keyword and a specified number of replicas there.

Database

Many things are missing in this service. So, let's go through using our troubleshooting steps, step by step.

- We see that there is no connectivity for the database service as there is no `networks` keyword specified.
- Port mapping is not required as we are not accessing the database from the host.
- We need to mount the volume as we need to pass our init script to Docker's `initdb.d` folder.
- Next, our Flask app will be exiting frequently since the database is not configured accordingly. So, pass env file to start the database with the configuration used in the Flask app.

This will fix all the issues with the database service.

Visualizer

In the visualizer service,

- Network connectivity is not established using `networks`. So, add `networks`.

That's it. There are no more issues here.

After fixing all these issues, hit the run button or deploy the stack if you are running it on a local machine.

You will be able to access the visualizer which will show you a continuous state of swarm cluster.

```
CREATE TABLE IF NOT EXISTS `users` (  
  `user_id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(255) DEFAULT NULL,  
  `password` varchar(50) DEFAULT NULL,  
  PRIMARY KEY (`user_id`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=10001 ;  
  
INSERT INTO `users` (`user_id`, `username`, `password`) VALUES  
(1, 'admin', 'admin123');
```

With this last exercise, we are at the end of this course and I think this must have been great learning for you. I suggest you go through the bonus section as well and not skip it. The bonus section will give you an idea of how Python applications are deployed in production systems. There are many ways to deploy applications in production and this is not the only way but the framework will be the same.

For any queries, you can reach out to me in the discussion forums or email me. So, I wish you all the best in your new journey where you will use your new skill and come out as version 2.0 of yourself.