

Configuration Manual

MSc Research Project Cloud Computing

Sai Dhawanjewar Student ID: 22130063

School of Computing National College of Ireland

Supervisor: Dr. Giovani Estrada

National College of Ireland

MSc Project Submission Sheet



School of Computing

Student Name:	Sai Dhawanjewar			
Student ID:	22130063			
Programme:	MSc Cloud Computing		Year:	2023
Module:	MSc Research Project			
Lecturer:	Dr. Giovani Estrada			
Due Date:	05/04/2024			
Project Title:	Improving Fault Tolerance a Microservices with a tempor	nd Consistency i ary DB	n	
Word Count:	2205	Page Count: 1	9	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

<u>ALL</u> internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sai Dhawanjewar

Date: 01/04/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only

onnee ose only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sai Dhawanjewar Student ID: 22130063

1 Introduction

This manual describes how to set up a test environment and run fault-tolerant microservices in Docker. It provides instructions for academics and student on how to install the required software and run the research project.

2 Required Software Tools

The following software tools are required for the project's execution:

Tool	Version	Use
Docker	24.0.6	Containerization platform for deploying and managing applications (Docker, 2024)
VS Code	1.85.1	Integrated development environment for developing microservices and React applications (Code, 2024)
Browser	Chrome: 120.0 Firefox: 120.0	Web browsers for testing and debugging the UI application (Chrome, 2024) (Firefox, 2024)
Python	3.11.1	The programming language used for the development of microservices-based applications (Python, 2024)
Django	3.2.18	A high-level Python web framework utilized for building microservices-based applications with efficiency and rapid development (Django, 2024)
React JS	18.2.0	JavaScript library for efficient development of dynamic, component-based web interfaces (ReactJs, 2024)
MySQL	Mysqlclient: 2.0.1 django-mysql: 3.9	A relational database management system used for storing and managing data in microservices-based applications (MySQL, 2024)
Git bash	2.43.0	A version-controlled collaborative space facilitating seamless cooperation, code transparency, and parallel development for efficient project management. (Git, 2024)

Note : Please follow the specific instructions for your operating system to install the prerequisites, as

these requirements were installed and run on Windows 11

Hardware Requirements

- Operating system: Windows/ Mac/ Linux (We can use any OS).
- RAM: Minimum 8GB / 16GB or more than needed.
- Processor: Any Processor from the Intel Core is suitable.

3 Prerequisite installation

The steps for the installation of software and tools are shown below.

1. Docker

Step1: Download Docker from official website (Docker, 2024): https://docs.docker.com/desktop/install/

Step 2: Click on button "Docker Desktop for windows". It will start downloading docker.

Step 3: Run the installer you downloaded.

Step 4: Follow the installation wizard's instructions.

Step 5: During installation, Docker Desktop may ask to enable Hyper-V and Windows features. Allow these changes if prompted.

Step 6: Launch Docker Desktop: Once the installation is complete, you can launch Docker Desktop from the Start menu.

Step 7: You may need to sign in using your Docker Hub account. If you don't have one, you can create it during the installation process.

Step 8: Docker Desktop will configure itself and start the necessary services. This might take a few minutes.

Step 9: Check Docker Version: Open a command prompt or PowerShell window. Type docker version and press Enter. This command should display information about your Docker installation.

2. VS Code

Step 1: Download Visual Studio Code (Code, 2024): <u>https://code.visualstudio.com/download</u>, Click on the "Download for Windows" button.

Step 2: Once the download is complete, run the installer (the .exe file you downloaded).

Step 3: Follow the instructions in the installation wizard. It may ask you to accept the license agreement and choose installation options.

Step 4: Click the "Install" button to start the installation process.

Step 5: Once the installation is complete, you can launch Visual Studio Code from the Start menu or desktop shortcut.

3. Browser

Step 1: Visit the official Google Chrome website (Chrome, 2024): https://www.google.com/chrome/

Step 2: Click on the "Download Chrome" button.

Step 3: The website will automatically detect your operating system and provide the appropriate download link.

Step 4: Click on the download link to start the download.

Step 5: Once the download is complete, run the installer.

Step 6: Follow the on-screen instructions in the installation wizard.

Step 7: Chrome will be installed on your system, and you can launch it from the Start menu or desktop shortcut.

Note : Follow same procedure to install firefox: https://mozilla-firefox.en.softonic.com/download

4. Python

Step 1: Download Python (Python, 2024): <u>https://www.python.org/downloads/</u>, Click on the "Downloads" tab.

You'll see the latest version of Python. choose between Python 3.x and Python 2.x. It is strongly recommended to use Python 3 as Python 2 has reached its end-of-life.

Step 2: Click on the download link for the installer corresponding to your Windows architecture (32-bit or 64-bit). Most modern Windows installations are 64-bit.

Step 3: Once the installer is downloaded, run it by double-clicking on the downloaded file

Step 4: Installation Wizard:

- Check the box that says "Add Python x.x to PATH" during the installation. This makes it easier to run Python from the command line.
- You can choose the "Customize installation" option to customize the installation location, but the default settings are usually fine for most users.

• Click on "Install Now" to start the installation.

Step5: Verify Python Installation:

- Open a command prompt or PowerShell window.
- Type python --version or python -V and press Enter. This should display the installed Python version.



If you see the Python version, it means the installation was successful.

5. Django

Step 1: Open 'cmd' or 'powershell' or 'Win+R' and type command – "pip install django" and press enter. This command will use the Python package manager (pip) to download and install the latest version of Django (Django, 2024).

Step 2: After the installation is complete, you can verify that Django is installed by checking its version. In the command prompt or PowerShell, type: python -m django –version.



Step 3: Navigate to the Project Directory with command – "cd projectname"

Step 4: Run the Development Server: To test the installation, run the following command to start the Django development server – "python manage.py runserver". This will start the development server, and you should be able to access your Django application by visiting http://127.0.0.1:8000/ in your web browser.

6. React JS

Step 1: Install Node.js and npm (ReactJs, 2024): https://nodejs.org/en/download/current

- Download the LTS (Long Term Support) version, which is recommended for most users.
- Run the installer and follow the installation instructions.

Step 2: Verify Installation: Open a command prompt or terminal window.Type the following commands to check if Node.js and npm are installed: node -v, npm -v

These commands should display the installed Node.js and npm versions.



Step 3: Create a React App: Install it globally using npm: "npm install -g create-react-app"

Step 4: Create a New React App:

- In the command prompt or terminal, navigate to the directory where you want to create your React app.
- Run the following command to create a new React app: "npx create-react-app my-react-app"

Step 5: Navigate to the App Directory: Change your current directory to the newly created app directory: "cd my-react-app"

Step 6: Run the Development Server: Start the development server with the following command: "npm start"

This will launch your React app, and you can view it by visiting http://localhost:3000 in your web browser.



4 Running proposed solution

Step 1: Start Docker engine.

Click on 'Docker Desktop' application \rightarrow It will open Docker application. \rightarrow Then sign in with your credentials.







Step 3: Open project "2Phase_impliment-master".

Click on 'Files' \rightarrow Click on 'Open folder' \rightarrow Select project folder '2Phase_impliment-master'



Step 4: Configure the IP address according to the internet connectivity for both the 'admin' and 'main1' services, as they will be required for API access in the project.

• First check available IP's. Go to command prompt \rightarrow type command 'ipconfig'

Command Prompt × + ·
Microsoft Windows [Version 10.0.22631.3007] (c) Microsoft Corporation. All rights reserved.
C:\Users\saidh>ipconfig
Windows IP Configuration
Ethernet adapter Ethernet 2:
Media State Media disconnected Connection-specific DNS Suffix . :
Unknown adapter Local Area Connection:
Media State Media disconnected Connection-specific DNS Suffix . :
Ethernet adapter Ethernet 3:
Connection-specific DNS Suffix .: Link-local IPv6 Address : fe80::42ff:361b:81bb:8ee8%8 IPv4 Address : 192.168.56.1 Subnet Mask : 255.255.255.0 Default Gateway :
Wireless LAN adapter Local Area Connection* 1:
Media State Media disconnected Connection-specific DNS Suffix . :

It will show you all available IP's.

- Here using IPv4 192.168.0.208 which is available for LAN adapter wifi. You can change it as per your connectivity.
- Set this IP for all services.
- Go to VS code. → select microservice 'admin' → select app 'admin' → select file 'settings.py'
- Copy IP added in Allowed host other than local host
 ALLOWED_HOSTS = ['localhost', '127.0.0.1', '192.168.173.181','0.0.0.0'] # Application definition
- Replace it with new IP selected as per your LAN availability.
- To replace IP same time at all locations click on \checkmark . Then enter old IP in first box and new IP in second box. And click on replace all button



Step 4: Deploy microservice "admin" on docker.

• Right click on microservice 'admin.' → select option 'Open in Integrated Terminal.'

×1 F	File Edit Selection	View Go	Run ····			✓ 2Phase_impliment-master	
СŊ	EXPLORER		🗬 settings.py	🛢 main1.sessi	on.sql	MySQLConnection.session.sql	≡ requirements.txt ×
	∨ 2PHASE_IMPLIMEN	T-MASTER	2Phase_implimer	nt-master > main1	> ≣ requ	irements.txt	
Q	✓ 2Phase_implim	ent-master	1 Django	==3.1.3			
/-	> .vscode		2 django	prestframework=	==3.12.2		
90	∽ admin		3 mysqlo	lient==2.0.1			
б	> .dbdata	New File			350		
\sim	> admin	New Folder.			5.5.0		
£	> product	Reveal in Fil	e Explorer	Shift+Alt+R			
-0	🕏 consumei	Open in Int	arated Terminal				
Щ	🕏 consumer	оренини					
_	≡ db.sqlite3	Find in Fold	er	Shift+Alt+F	/ pip t	to install required python pac	kages
L_⊚	👉 docker-cc					in emerics. txt	
	🔷 Dockerfile	Cut		Ctrl+X			
A	🗬 manage.p	Сору		Ctrl+C			
	≣ requirem∉			Ctrl+V			
-	> Frontend	C D H					
	> main1	Copy Path		Shift+Alt+C	.4.4		
Π	MySQLCon	Copy Relativ	ve Path Ctrl	+K Ctrl+Shift+C			
0		Rename		F2			
		Doloto		Dolote			
		Delete		Delete			
			25 # pika 24	==1.1.0			

• It will open the terminal window.



• Now enter the command 'docker-compose up –build' to deploy microservice 'admin' on docker.

PS	D:\Reserch project\2Phase impliment-master\2Phase impliment-master\admin> docker-compose upbuild	
[+]	Building 38.5s (14/17) docker:	default
	[backend internal] load build definition from Dockerfile	0.0s
	=> transferring dockerfile: 229B	0.0s
	[backend internal] load .dockerignore	0.0s
	=> transferring context: 2B	0.0s
	[queue internal] load .dockerignore	0.0s
	=> transferring context: 2B	0.0s
	[queue internal] load build definition from Dockerfile	0.0s
	=> transferring dockerfile: 229B	0.0s
	[queue internal] load metadata for docker.io/library/python:3.9	2.3s
	[backend 1/5] FROM docker.io/library/python:3.9@sha256:3d9dbe78e1f45ed2eb525b462cdb02247cc0956713325aeeffa37cb5f2c8c	23.4s

Step 5: Verify the deployed service 'admin' on docker.

Docker Desktop Update to latest	Q Search for imag	es, containers, volumes, extensions a	nd more Ctrl+K		ð 🌣 Sign	in 🛛 🚽 🗇 🗡
Containers	Containers Give feedback 🗣					
Images Volumes Dev Environments BETA	Container CPU usage ① 0.12% / 1000% (10 cores allocated)		Container memory usag 232.27MB / 3.59G	Container memory usage ① 232.27MB / 3.59GB		Show charts 🗸
Docker Scout	Q, Search	Only show running	g containers			
Learning center	Name	Image	Status	CPU (%) Port(s)	Last started	Actions
Extensions :	🗆 👻 📚 admin		Running (3/3)	0.12%	in 0 seconds	$\mathbf{x}_{i} \in \{1, 2, \dots, N\}$
Add Extensions	□	mysql:5.7.22	Running	0.12% <u>33066:3306</u>	2 seconds ago	$\mathbf{x}_{i} \in [1,\infty)$
	backend-1 bbfddb127681	admin-backend	Running	0% <u>8003:8003</u> 🛛	in 0 seconds	$\mathbf{x}_{i} \in \{1, 2\}$
	ea6aeea12187	admin-queue	Running	0%	0 seconds ago	* 1 I *

Here we can see 'admin' microservice is deployed successfully. And it is Up and Running.

Step 6: Follow the same procedure to deploy the microservice 'main1' as 'admin' on Docker.



Step 7: Verify the deployed service 'main1' on docker.

Docker Desktop Update to latest		Q Search for images, contain	ners, volumes, extensions and more	Ctrl+K		ð 🌣 Sign	in \varTheta	=	a ×
Containers	Containers Give for	eedback 🖼							
 Images Volumes Dev Environments BETA 	Container CPU usage (3.01% / 1000% (10 d	D cores allocated)	Container memory usag 361.63MB / 3.59G		Ū		Show charts 🗸		~
Docker Scout	Q Search		Only show running containe	rs					
S Learning center	Name Name	In	nage	Status	CPU (%) Port(s)	Last started	Actions		
Extensions I	🗆 👻 😂 ad	lmin		Running (3/3)	1.56%	12 minutes ago	(e) - 2		
Add Extensions		db-1 132967ccfd06 🗊 🛄	<u>iysql:5.7.22</u>	Running	0.07% <u>33066:3306</u>	12 minutes ago) .		
		bbfddb127681	dmin-backend	Running	1.49% <u>8003:8003</u>	12 minutes ago		1	
		gueue-1 ea6aeea12187 ₪ ac	dmin-queue	Running	0%	12 minutes ago			
	🗆 - 😂 ma	ain1		Running (3/3)	1.45%	7 minutes ago		:	
		db-1 53b8250333b9 € m	<u>iysql:5.7.22</u>	Running	0.08% <u>33068:3306</u>	7 minutes ago	•	:	
		queue-1 5f4afa428853 ₪ m	ain1-queue	Running	0%	7 minutes ago			
		backend-1 ebcf53f507c1	ain1-backend	Running	1.37% <u>8002:8002</u>	7 minutes ago	•	:	
							Sh	owing 8	items
🐣 Engine running 🛛 🕨 🔢 🕖 🗄	RAM 1.80 GB CPU 0.35%	🕷 Not signed in					C	v4.24.2	.2

Here we can see 'main1' microservice is deployed successfully. And it is Up and Running.

Step 8: Start 'Frontend' application.

- Edit Selection View Go ₽ 2Phase_impliment-master Run main1.session.sql MySQLConnection.session.sql EXPLORER settings.py \equiv requirements.txt \times ſĴ ✓ 2PHASE_IMPLIMENT-MASTER 2Phase_impliment-master > main1 > ≡ requirements.txt Django==3.1.3 Q ✓ 2Phase_impliment-master djangorestframework==3.12.2 > .vscode mysqlclient==2.0.1 > admin ço django-mysql==3.9 ✓ Frontend s==3.5.0 > node New File... > publ New Folder... > src Reveal in File Explorer Shift+Alt+R ₽ .gitic Open in Integrated Terminal {} pack -_0 {} pack Find in Folder... Shift+Alt+F REAL Cut Ctrl+X 囚 🖪 tscor > main1 Сору Ctrl+C 4-01-19T21:22:47.418454Z 0 [Warning] 'user' entry 'mysql.sys@localhe MySQ 4-01-19T21:22:47.419885Z 0 [Warning] 'db' entry 'performance_schema e mode. Copy Path Shift+Alt+C 4-01-19T21:22:47.419940Z 0 [Warning] 'db' entry 'sys mysql.sys@loca ٢ **Copy Relative Path** Ctrl+K Ctrl+Shift+C
- Open terminal for react application 'Frontend.'

- Verify if 'npm' command is working at desired location or not. Otherwise install npm with command- '**npm install**'
- Then start npm with command 'npm start'. This will start the react application at

port http://localhost:3000, http://192.168.56.1:3000



Step 9: Open application in desired browser.

Open Admin page with URL - 'http://localhost:3000/admin/products'





Open Main page with URL - 'http://localhost:3000/'

5 Used case execution with result

1 Normal flow without failure

Use Case 1 explains the normal or expected flow when a user attempts to add, update, retrieve, or delete a product through the Admin page and tries to retrieve and like products through the Main page. Following are the steps to execute the scenarios.

• Step 1: To Update image in web application. Access the Admin page with URL - 'http://localhost:3000/admin/products' in the preferred browser



• Step 2: Then click on the 'Edit' button. This will open the Edit page. Then update the title and click on the save button.

$\leftarrow \rightarrow C$	0 0	localhost:3000/admin/products/48/edit	110%	ŝ
My ArtGallery				
Products				
		Title		
		Image updated		
		Image		
		https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSV1rARdKWopb7On1bDS5hH5_7bjT7ZH1n1U-Fb-ysCpA	&s	
		Save		

• Step 3: This will update the image title for the selected image.



• Step 5: To check data consistency. Open Main page with URL – 'http://localhost:3000/' and check Same image is added on it.



• Step 6: In this manner, data consistency is maintained in the normal or regular flow. To verify data consistency for scenarios involving Add, Delete, and Like operations, one can follow the same steps and examine the results.

2 Fault flow with 2PC, saga and temporary DB

Use Case 2 Outlines the fault-handling mechanism when any of the microservices is not operational. It illustrates how the Two-Phase Commit and Saga patterns collaborate with a temporary database during the recovery process.

110% 🖒

٢

• Step 1: Normal flow

0

Step 1: Open main page with URL - 'http://localhost:3000/'



Step 2: Click on the like button for any product, check the incremented count. Verify 0 incremented like count.



• Step 3: Open Admin page and check like count for the same image by refreshing Admin page.

My Attallay 4 4 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	$\leftarrow \rightarrow $ C	00	localhost:3000/admin/p	products								110% ర్	2	C	9 3 2) =
A dd formions Image: participant Image: participant <td>My ArtGallery</td> <td></td>	My ArtGallery															
A discussion A discussion 0 is provided Image: packed 1 0 Image: packed 1 1 Image: packed														_		
I arrive a second apper a second			43						The	temple bar			0	Edit	Delete	
A displayed for the product is matched and data consistency is according to the product is matched and data consistency is according to the product is matched and data consistency is according to the product and fault occurs. • • • • • • • • • • • • • • • • • • •			WIN.	THE TEMP	E BAR											
I define the fact			<u>Ab</u>		HA											
Cupul, Image Cupul, Image I tell parties I																
 a terming containers b containers c containers <			47		- Andrew				Orig	ginal_Image		l	1	Edit	Delete	
a general g			A.													
a property of the second se			Prolote.	la												
 a gradie gradied b range updated b range updated c rank r ran			a setter		. M.M.											
Add Extensions Add Extensions Status CPU(%) Port(s) Last started Actions Container <p< td=""><td></td><td></td><td>48</td><td>_</td><td>-</td><td></td><td></td><td></td><td>Ima</td><td>ge updated</td><td></td><td></td><td>5</td><td>Edit</td><td>Delete</td><td></td></p<>			48	_	-				Ima	ge updated			5	Edit	Delete	
 Step 4: Like count displayed for the product is matched and data consistency is achieved in normal. Step 2: Fault occurred: Now user tries to Like some product and fault occurs. Step 1: To create the fault scenario, turn of the backend service of Admin in docker Container Succedback Volumes Volumes Container Succedback Container CPU usage Container Succedback Container Succedback Container CPU usage Containers Container Succedback Status CPU (W) Port(s) Last started Actions Seerch I Container Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 1.87% 8022802 g 58 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i i Succedback Running 0.33% 59 seconds ago i i i Running 0.33% 59 seconds ago i				0												
 Step 4: Like count displayed for the product is matched and data consistency is achieved in normal. Step 2: Faut occurred: Now user tries to Like some product and fault occurs. Step 1: To create the fault scenario, turn of the backend service of Admin in docker Containers Containers Container				7												
 Step 4: Like count displayed for the product is matched and data consistency is achieved in normal. Step 2: Fault occurred: Now user tries to Like some product and fault occurs. Step 1: To create the fault scenario, turn of the backend service of Admin in docker Containers @xcfeedback@ Container CPU usage O Container CPU usage O Container CPU usage O Show charts Container CPU usage O Status CPU (%) Port(s) Lest started Actions Step 1: To create the fault scenario, turn of normal. Container CPU usage O Show charts Container CPU usage O Show charts Container CPU usage O Status CPU (%) Port(s) Lest started Actions Status OPU (%) 0.03% 59 seconds ago i i i i i i i i i i i i i i i i i i i	- 6	C4.000	A. Lilve e		diamla	und for	41	dar of			به المعر ال			:		
 Step 2: Fault occurred: Now user tries to Like some product and fault occurs. Step 1: To create the fault scenario, turn of the backend service of Admin in docker Containers Gue Ledback % Volumes Volumes Dev Environments arx Decker Scout Learning center Mane mage Status CPU (%) Port(s) Last started Actions Gesegoes 2000 (1) (1) (2) (2) (2) (2) (2) (2) (2) (2) (2) (2	02	step achie	eved in no	ount ormal	dispia	yed for	the pro	auci	. 15 1	matche		a consisi	ency	15		
 Step 1: To create the fault scenario, turn of the backend service of Admin in docker Containers Size feedback % Volumes Volumes Docker Scout Search III Only show running containers Search III Only show running containers Search III Only show running containers Search III Only show running 0.03% 59 seconds ago III Only show IIII Saskend Running 1.87% 8002.8002 (2) 58 seconds ago III III Search III Saskend Running 1.87% 8002.8002 (2) 58 seconds ago III III Search IIII Saskend Running 1.87% 8002.8002 (2) 58 seconds ago III IIII Search IIII Saskend Running 1.87% 8002.8002 (2) 58 seconds ago III IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	• Step 2: Fa	ault	occurred	l: No	w use	r tries to	b Like s	some	e pr	oduct a	and fault o	occurs.				
Containers Containers Containers Gue feedback © Volumes Container CPU usage () Stass definer Show charts v Container CPU usage Stass definer Deck Environments serat Decker Scout Search Image Status Centrainers Gue Search Image Status Centrainers Gue Search Centrainers Gue Search Image Status Centrainers Gue Search Image Status Centrainers Gue Search Image Status Centrainers Gue Search Image <td>0 \$</td> <td>Step</td> <td>1: To cre</td> <td>ate th</td> <td>ne faul</td> <td>lt scenar</td> <td>rio, tur</td> <td>n of</td> <td>the</td> <td>backer</td> <td>nd service</td> <td>of Adn</td> <td>nin in</td> <td>dock</td> <td>er</td> <td></td>	0 \$	Step	1: To cre	ate th	ne faul	lt scenar	rio, tur	n of	the	backer	nd service	of Adn	nin in	dock	er	
Images Volumes Dev Environments serA Container CPU usage ○ Container CPU usage ○ Sta3.34MB / 3.59GB Dev Environments serA Container CPU usage ○ Container CPU usage ○ Sta3.34MB / 3.59GB Container CPU usage ○ Container CPU usage ○ Sta3.34MB / 3.59GB Container CPU usage ○ Container CPU usage ○ Sta3.34MB / 3.59GB Container CPU usage ○ Container CPU usage ○ Sta3.34MB / 3.59GB Container CPU usage ○ Container CPU usage ○ Sta3.34MB / 3.59GB Container CPU usage ○ Container CPU usage ○ Container selected Container CPU usage ○ Container CPU usage ○ Container selected Container CPU usage ○ Container CPU usage ○ Container selected Container CPU usage ○ Container CPU usage ○ Container selected Container CPU usage ○ Container containers Cont	Containers		Container	S Give	eedback 🖳											
Volumes 2.16% / 1000% (t0 cores allocated) 543.34MB / 3.59GB Dev Environments BETA Q Search III ● Only show running containers Learning center III ● Only show running containers Extensions III ● Only show running 0.03% 59 seconds ago Image Status CPU (%) Port(s) Last started Actions Image Image Running 0.03% S9 seconds ago Image Image Image admin-backend Exited 0%	💮 Images		Container CP	Uusage (Ð					Container mer	mory usage (i)		Sh	ow chart	s 🗸	
Image Status CPU (%) Port(s) Last started Actions Image Image Running 0.03% 59 seconds ago Image Image Image Image Image Exited 0%<	Volumes		2.16% / 10	100% (10	cores allocat	ed)				543.34MB	/ 3.59GB					
● Docker Scout ● Other Image Status CPU (%) Port(s) Last started Actions ■ Extensions ■ G6a959c36 ■ G6a95	Dev Environments BETA	A	Q Search	1		ш	Only	show r	unninc	1 containers						
Learning center Image Status CPU (%) Port(s) Last started Actions Image Status CPU (%) Port(s) Last started Actions Image Status Image Image </td <td>Docker Scout</td> <td></td> <td>- Cocaron</td> <td></td> <td></td> <td></td> <td>• • • • • • • • • • • • • • • • • • •</td> <td>CHOWIN</td> <td>unnig</td> <td>goontamero</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	Docker Scout		- Cocaron				• • • • • • • • • • • • • • • • • • •	CHOWIN	unnig	goontamero						
Extensions : Add Extensions Add Exten	Learning center			Name		Image	Status			CPU (%)	Port(s)	Last started	Acti	ons		
Add Extensions <li< td=""><td>Extensions</td><td>:</td><td>_</td><td></td><td>66a959c36</td><td></td><td></td><td>3</td><td></td><td>0</td><td></td><td></td><td></td><td></td><td>-</td><td></td></li<>	Extensions	:	_		66a959c36			3		0					-	
Image: Second sequence of the second	Add Extensions				5c3908578	<u>main1-queue</u>	Runnir	,g		0.03%		59 seconds	ago 🔳	1	Ĩ	
Image: Seconds ago Image: Se	-				backend- 2f01d2ba1	main1-backen	<u>d</u> Runnir	ıg		1.87%	8002:8002	58 seconds	ago 🔳	:	Î	
Image: state of the state			□ ·	<mark>ම</mark> ac	<u>lmin</u>		Runnir	ıg (2/3)		0.14%		58 seconds	ago 🔳	÷		
Image: Strateging admin-backend Exited 0% 8003:8003 IZ 58 seconds ago Image: Strateging admin-backend Image: Strateging admin-backend Exited 0% 8003:8003 IZ 58 seconds ago Image: Strateging admin-backend Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue Image: Strateging admin-gueue Running 0.03% 59 seconds ago Image: Strateging admin-gueue <					<u>db-1</u>	mysql:5.7.22	Runnir	ıg		0.11%	<u>33066:3306</u> 🛛	1 minute ago		I	ĩ	
Image: State of the state					backend-	admin-backen	d Exited			0%	8003:8003 17	58 seconds	ado 🕨	1		
e800f611d admin-queue Running 0.03% 59 seconds ago : Showing 8 items			Ľ	-	3155ef803 <u>queue-1</u>								-3-			
Showing 8 items					e800f611d	admin-queue	Runnir	g		0.03%		59 seconds	ago	:		
Engine running Number 1 RAM 3 67 GB CPU 11 86% Not signed in	de Engine running Dut	(b) ;]	RAM 3 67 CR CP	111.96%	Not sign	ed in								Showin		2

- Step 3: Fault handling flow with 2PC -
 - **Step 1:** Try to like any product on the Main page, this will not allow Main service to change the like count as Admin service is not reachable.



- Step 3: Recovery mechanism with Saga pattern -
 - Step 1: To check recovery scenario turn on the backend service in docker. And wait until service comes up.

Containers	Containers Give feedback		
Images Volumes Pay Environments	Container CPU usage ① 35.09% / 1000% (10 cores allocated)	Container memory usage 🚯 617.8MB / 3.59GB	Show charts 🗸
Docker Scout	Q Search	Only show running containers	
Extensions	Name Image	Status CPU (%) Port(s)	Last started Actions
Add Extensions	backend- 2/01d/ba1 main1-backend	Running 0% Running 12.15% <u>8002:8002</u>	10 minutes ago E : E
	□ ~ \$ admin	Running (3/3) 22.81%	18 seconds ago 🔹 🗄
	4e84ae473 mysql:5.7.22	Running 10.63% 33066:3306	10 minutes ago 🔳 🚦 📋
	backend- 3155ef803 admin-backend	Running 12.18% <u>8003:8003</u>	18 seconds ago 🔳 🚦
	e800f611d	Running 0%	10 minutes ago 🔹 🗄 🗎

- Showing 8 items
- **Step 2:** With the implementation influenced by Saga pattern, Admin service will regain its original state and Main service will process the all the records available in temporary memory and will send it to Admin service.
- Step 4: To check the result achieved by failure recovery. Refresh the Main page check the like count for the product.



Step 5: Open the Admin page to check the like count of the same product which will be equal.
 ○ □ locathost3000/admin/products



References

Docker. (2024). Retrieved from https://www.docker.com/

Google Chrome. (2024). Retrieved from https://www.google.com/chrome/.

Mozilla Firefox. (2024). Retrieved from https://www.mozilla.org/en-US/firefox/

- Python. (2024). Retrieved from https://www.python.org/
- Django. (2024). Retrieved from https://www.djangoproject.com/
- React. (2024). Retrieved from https://reactjs.org/
- MySQL. (2024). Retrieved from https://www.mysql.com/
- Git for Windows. (2024). Retrieved from https://gitforwindows.org/