

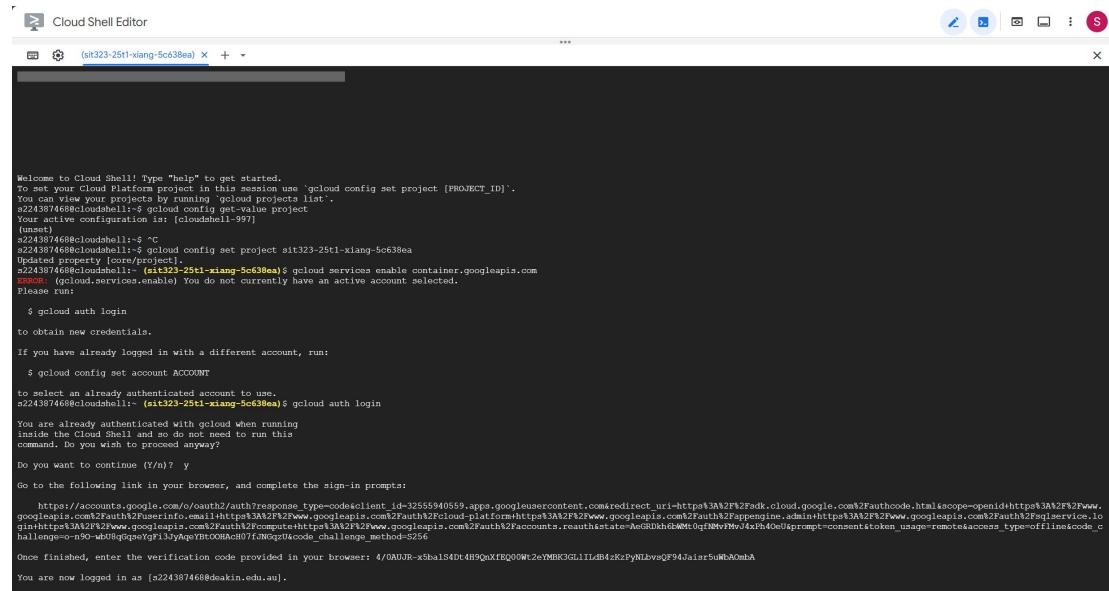
SIT323 11. 3HD

GitHub Link:

[xsc224387468/Deakin-Marketplace](https://github.com/xsc224387468/Deakin-Marketplace)

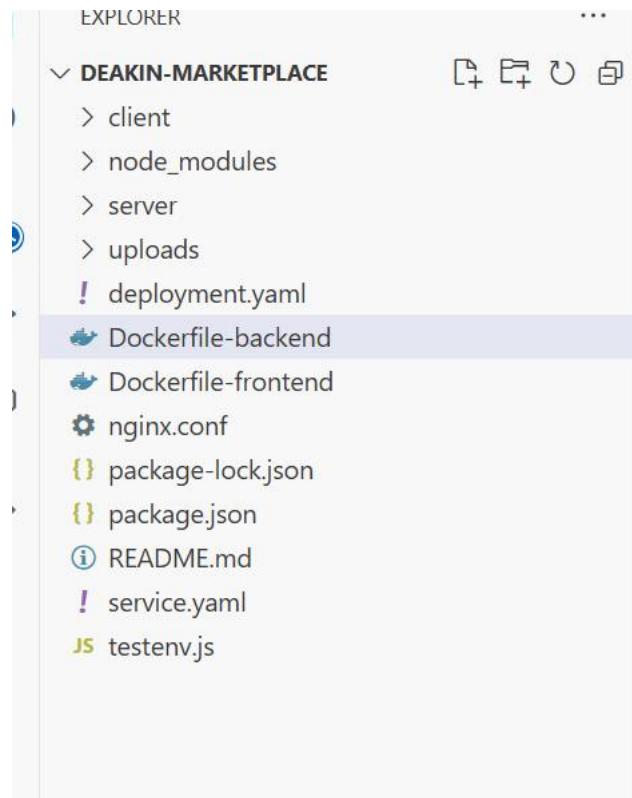
Deployment Process:

Connect to the cloud shell and download the project from GitHub
Log in to your gcloud account



Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use "gcloud config set project [PROJECT_ID]".
You can view your projects by running "gcloud projects list".
\$ gcloud config get-value project
Your active configuration is: [cloudshell-997]
(unset)
\$ gcloud config set project sit323-25t1-xiang-5c638ea
Updated property [core/project].
\$ gcloud services enable container.googleapis.com
ERROR: (gcloud.services.enable) You do not currently have an active account selected.
Please run:
\$ gcloud auth login
to obtain new credentials.
If you have already logged in with a different account, run:
\$ gcloud config set account ACCOUNT
to select an already authenticated account to use.
\$ gcloud auth login
You are already authenticated with gcloud when running
inside the Cloud Shell and so do not need to run this
command. Do you wish to proceed anyway?
Do you want to continue (Y/n)? y
Go to the following link in your browser, and complete the sign-in prompts:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=325590658486-googleapis.com%2Faauth%2Fwww.googleapis.com%2Faauth%2Fcloud-platform%2Fhttps%3A%2F%2Fwww.googleapis.com%2Faauth%2Fsign-in%2Faccounts%2Faccounts-state%3Dfb%26state%3Dfb%26code_challenge_method=S256&challenge_method=S256&code_challenge=qBqeyfI3yApeYHrO9A-H07fcmNGzgt2%code_challenge_method=S256
Once finished, enter the verification code provided in your browser: 4/0AUUR-x5ba1S4Dt4RH9QnXIEQ00Wt2eYMBK3GL1LlDb4zK2PyNLbvqQF34JaIsr5uRbR0mA
You are now logged in as [s224387468@deakin.edu.au].

Create Dockerfile, deployment, and service files



```
↳ Dockerfile-backend M X ↳ Dockerfile-frontend

↳ Dockerfile-backend
1 # Dockerfile-backend
2 FROM node:18
3
4 WORKDIR /app
5
6 # Copy all back-end files
7 COPY server/ ./server/
8 COPY uploads/ ./uploads/
9 COPY package*.json ./
10
11
12 RUN npm install
13
14 # port
15 EXPOSE 5000
16
17 # start server
18 CMD ["node", "server/index.js"]
19
```

```
↳ Dockerfile-backend M X ↳ Dockerfile-frontend

↳ Dockerfile-frontend
1 # Stage 1: Build React app
2 FROM node:18-alpine AS build
3
4 WORKDIR /app
5
6 # Copy only package files first for better caching
7 COPY client/package.json client/package-lock.json ./client/
8 WORKDIR /app/client
9 RUN npm install
10
11 # Copy all client files and build
12 COPY client/ ./
13 ENV REACT_APP_API_URL=http://34.129.60.74
14 RUN npm run build
15
```

```

15
16 # Stage 2: Serve with Nginx
17 FROM nginx:stable-alpine
18
19 # Add custom Nginx config to support React Router
20 COPY nginx.conf /etc/nginx/conf.d/default.conf
21
22 # Copy built frontend files to Nginx public directory
23 COPY --from=build /app/client/build /usr/share/nginx/html
24
25 EXPOSE 80
26 CMD ["nginx", "-g", "daemon off;"]
27

```

Link to Google Cloud Platform Project

```

You are now logged in as [s224387468@deakin.edu.au].
Your current project is [sit323-25tl-xiang-5c638ea]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
s224387468@cloudshell:~ (sit323-25tl-xiang-5c638ea)$ gcloud config set project sit323-25tl-xiang-5c638ea
Updated property [core/project].

```

Log in to Docker Hub

```

root@100-104-10-104:~# ssh s224387468@cloudshell -t /bin/bash
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker tag australia-southeast2-docker.pkg.dev/sit323-25tl-xiang-5c638ea/marketplace-repo/deakin-marketplace:v1 shichengxiang/deakin-mark
etplace:v1
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker login
Using WEB-BASED LOGIN
Info - To sign in with credentials on the command line, use 'docker login -u <username>'.

Your one-time device confirmation code is: DKCL-TDXZ
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate
Waiting for authentication in the browser.

'Clogin canceled
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker login -u shichengxiang
Info - A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://app.docker.com/settings

Password:
WARNING! Your credentials are stored unencrypted in '/home/s224387468/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/guide/credential-store/
Login Succeeded
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker tag australia-southeast2-docker.pkg.dev/sit323-25tl-xiang-5c638ea/marketplace-repo/deakin-marketplace:v1 xcdeckeruser/deakin-mark
etplace:v1

```

Build the front-end image and tag the version

```

root@100-104-10-104:~# ssh s224387468@cloudshell -t /bin/bash
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker build -t gcr.io/sit323-25tl-xiang-5c638ea/deakin-marketplace:v1 .
[+] Building 7.5s (8/9)
--> [internal] load build definition from Dockerfile
--> [internal] load metadata for docker.io/library/node:18
--> [internal] load .dockerignore
--> [internal] transfer context: 2B
--> CANCELED [1/5] FROM docker.io/library/node:18@sha256:867be0f97d45cb7d89a0ef0b328d23e8207412ebec456441ed8cabcc4ecd
--> > resolve docker.io/library/node:18@sha256:867be0f97d45cb7d89a0ef0b328d23e8207412ebec456441ed8cabcc4ecd
--> => sha256:639f6a05108254dc31ff8df9d0f6714c7a793aa0aa4ae3d7f4f46504ac8 24.01MB / 24.01MB
--> => sha256:867be0ff7d45c7d9a0ef0b328d23e8207412ebec456441ed8cabcc4ecd 6.41KB / 6.41KB
--> => sha256:a0ee20d21ff7e319f16644fd076ef1435b42689ca015daf08e47c02fd6ccf39 2.49KB / 2.49KB
--> => sha256:e55aee3bc5724c51df0a1ad86ce38aee233bb34345798c943c1c7c1c9a2a84 6.39KB / 6.39KB
--> => sha256:c0f05a52cd2353f0b2b6f9be0549ac91663fb1dc8d4bac405eacf/f28562e9f2 48.49MB / 48.49MB
--> => sha256:c513cad200b13ead2dc74549859ed58a6cb3480e3ba6117f854da097262526 64.39MB / 64.39MB
--> => extracting sha256:cfc05a52cd2353f0b2b6f9be0549ac91663fb1dc8d4bac405eacf/f28562e9f2
--> => sha256:c187b51b626el60ab369727b81f440adea9d45e97a45e137fc318be0bf7f09f 161.23MB / 211.36MB
--> => sha256:60b42icb1807599zb04ab52f93de1450cf10de1cf73idat9ea52af3dd008e42a 3.52KB / 3.52KB

```

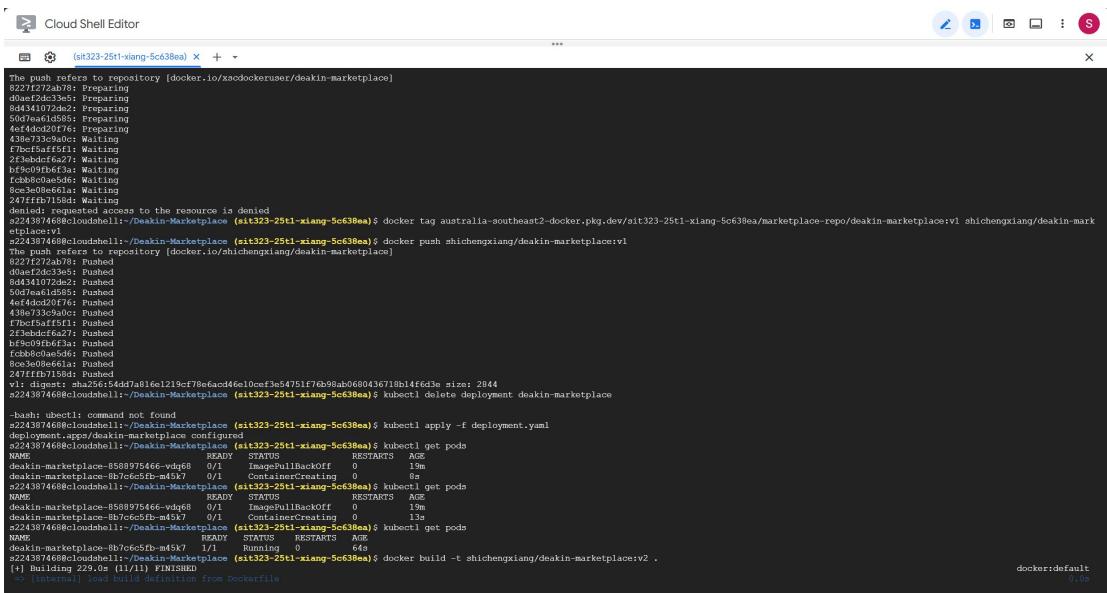
Push the built image to Docker Hub

```

Adding credentials for all GCR repositories.
WARNING: A long list of credential helpers may cause delays running 'docker build'. We recommend passing the registry name to configure only the registry you are using.
gcloud credential helpers are not recommended correctly.
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker push gcr.io/sit323-25tl-xiang-5c638ea/deakin-marketplace:v1
The push refers to repository [gcr.io/sit323-25tl-xiang-5c638ea/deakin-marketplace]
0227f272ab78: Retrying in 1 second
0227f272ab78: Retrying in 1 second
0d341072d2: Retrying in 1 second
50d7ea61d585: Retrying in 1 second
4ef4cd20f76: Retrying in 3 seconds
4386733c9a0c: Retrying in 4 seconds
f7bcf5aff5f51: Waiting
2f3ebdfdf6a27: Waiting
bf9c09f6f63a: Waiting
fcbb8c0ae5de: Waiting
8ce3e08661a: Waiting
247ff0b134: Waiting
manifest invalid: location "us" violates organization policy on the project sit323-25tl-xiang-5c638ea
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ gcloud services enable artifactregistry.googleapis.com
s224387468@cloudshell:~/Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ gcloud artifacts repositories create marketplace-repo \
--repository-format=docker \
--location=australia-southeast2 \
--description="Docker repo for SIT323 project"
Create request issued for: [marketplace-repo]
Waiting for operation [projects/sit323-25tl-xiang-5c638ea/locations/australia-southeast2/operations/897c511d-c678-451a-8alc-bdbad272dd3] to complete...done.
Created repository [marketplace-repo].

```

- Deploy or update the frontend to GKE
kubectl apply -f client/deployment-frontend.yaml
- Create or update a backend LoadBalancer service
kubectl apply -f service.yaml
- Check the status of the current Pod to confirm whether it is running
kubectl get pods
- View service information and obtain the EXTERNAL-IP of the front-end and back-end
kubectl get services
- View the log output of the backend service to troubleshoot operation problems
kubectl logs deployment/deakin-marketplace

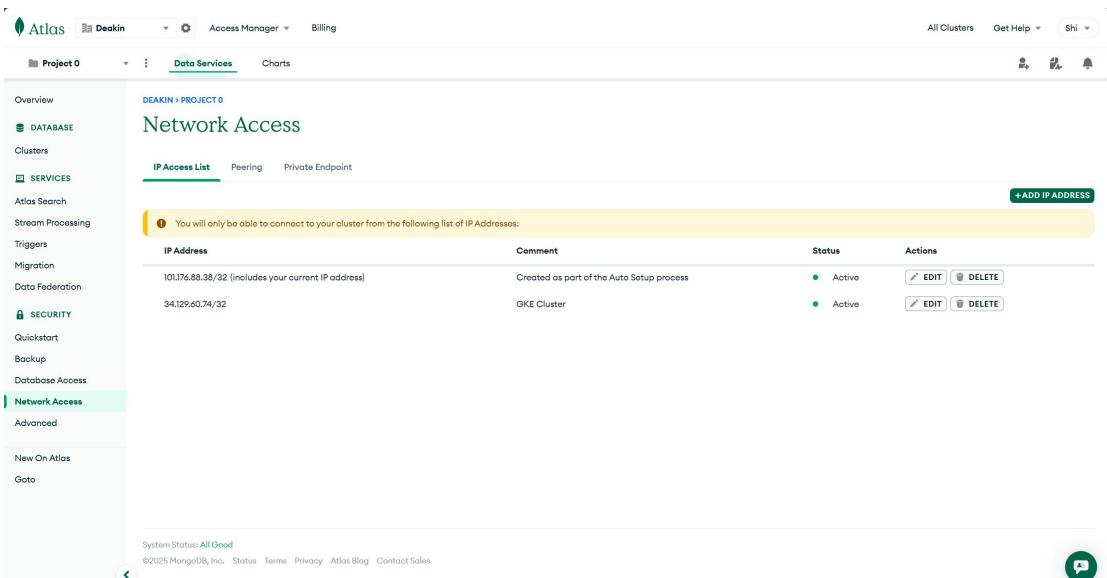


```

Cloud Shell Editor
(git323-25t1-xiang-5c638ea) + ...
The push refers to repository [docker.io/xsclockruser/deakin-marketplace]
8227f272ad78: Preparing
d0aef02a010d: Preparing
8d4341072de0: Preparing
50d7eaf1d585: Preparing
4effd0e20103: Preparing
4fb9c05e05c0: Preparing
f7bcff5aff5f1: Waiting
2f3ebd5fe5f1: Waiting
4fb9c05e05c0: Waiting
fcbb80ae5d5d: Waiting
8ce3a09e616a: Waiting
247fffb7158d1: Waiting
d0aef02a0103: Waiting access to the resource is denied
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ docker tag australia-southeast2-docker.pkg.dev/sit323-25t1-xiang-5c638ea/marketplace-repo/deakin-marketplace:v1 shichengxiang/deakin-marketplace:v1
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ docker push shichengxiang/deakin-marketplace:v1
The push refers to repository [docker.io/shichengxiang/deakin-marketplace]
8227f272ad78: Pushed
d0aef02a0103: Pushed
8d4341072de0: Pushed
50d7eaf1d585: Pushed
4effd0e20103: Pushed
4fb9c05e05c0: Pushed
v1: digest: sha256:54dd7a816e1219cf78e6acd46e10cef3e54751f76b98ab06804367181bf6d3e size: 2844
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ kubectl delete deployment deakin-marketplace
+-----+
-bash: ubectl: command not found
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ kubectl apply -f deployment.yaml
deployment.apps/deakin-marketplace created
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deakin-marketplace-5b7fc5fb-m45k7   1/1   ImagePullError   0   4m
deakin-marketplace-5b7fc5fb-m45k7   0/1   ContainerCreating   0   8s
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deakin-marketplace-5b7fc5fb-m45k7   1/1   ImagePullError   0   11s
deakin-marketplace-5b7fc5fb-m45k7   0/1   ContainerCreating   0   11s
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deakin-marketplace-5b7fc5fb-m45k7   1/1   Running   0   49s
a224387464@cloudshell:~/Deakin-Marketplace (git323-25t1-xiang-5c638ea)$ docker build -t shichengxiang/deakin-marketplace:v2 .
[+] Building 229.0s (11/11) FINISHED
--> [internal] load build definition from Dockerfile
docker:default 0.0s

```

Enable MongoDB to allow backend domain name access



The screenshot shows the MongoDB Atlas Network Access page for the 'DEAKIN > PROJECT0' cluster. Under the 'IP Access List' tab, there is one entry for the current IP address (10.176.88.38/32) with a comment 'Created as part of the Auto Setup process'. Another entry for the GKE Cluster (34.129.60.74/32) is also listed with a comment 'GKE Cluster'. Both entries have an 'Active' status and edit/delete buttons.

IP Address	Comment	Status	Actions
10.176.88.38/32 (includes your current IP address)	Created as part of the Auto Setup process	Active	<button>EDIT</button> <button>DELETE</button>
34.129.60.74/32	GKE Cluster	Active	<button>EDIT</button> <button>DELETE</button>

Successfully connected to MongoDB

```

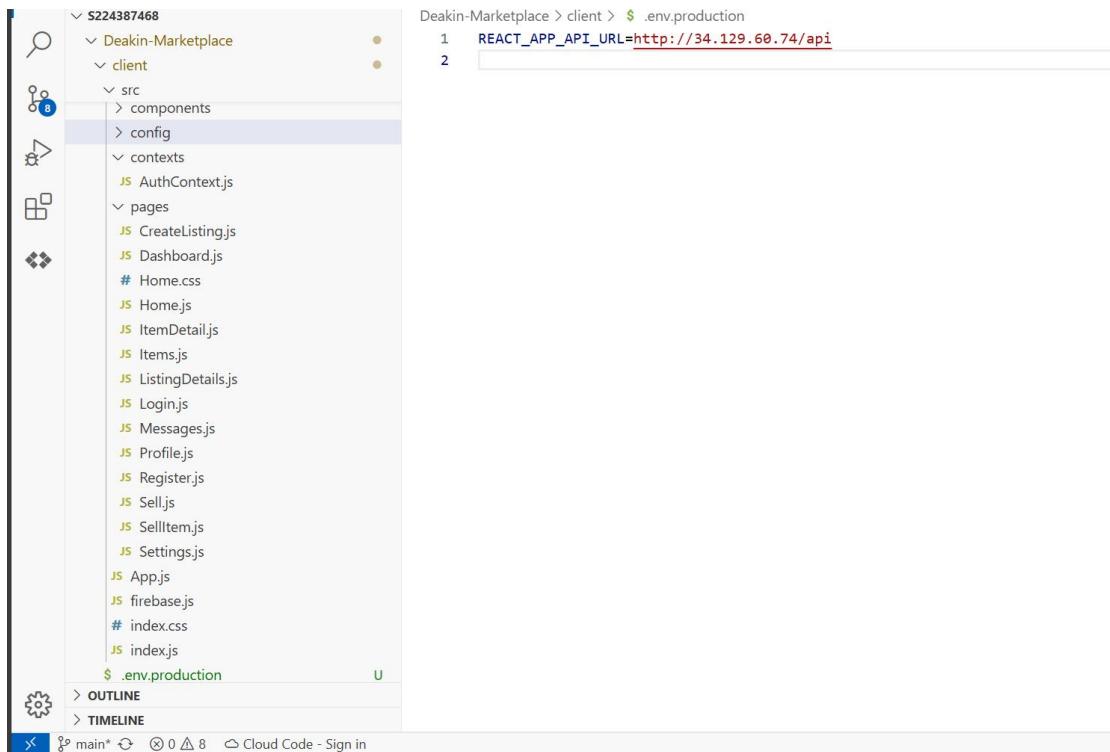
n224387468@cloudshell:~/Deakin-Marketplace$ (sit323-25t1-xiangg-5c638ea)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
deakin-marketplace-948f76c55-9gc6c   1/1     Running   0          96s
n224387468@cloudshell:~/Deakin-Marketplace$ (sit323-25t1-xiangg-5c638ea)$ kubectl logs -l app=deakin-marketplace
MongoDB connection error: Could not connect to any servers in your MongoDB Atlas cluster. One common reason is that you're trying to access the database from an IP that isn't whitelisted. Make sure your current IP address is on your atlas cluster's IP whitelist: https://www.mongodb.com/docs/atlas/security-whitelist/
IP whitelisting is on. Atlas client's IP is whitelisted.
MongoDB connection error: Could not connect to any servers in your MongoDB Atlas cluster. One common reason is that you're trying to access the database from an IP that isn't whitelisted. Make sure your current IP address is on your atlas cluster's IP whitelist: https://www.mongodb.com/docs/atlas/security-whitelist/
Retrying connection in 5 seconds...
MongoDB connection error: Could not connect to any servers in your MongoDB Atlas cluster. One common reason is that you're trying to access the database from an IP that isn't whitelisted. Make sure your current IP address is on your atlas cluster's IP whitelist: https://www.mongodb.com/docs/atlas/security-whitelist/
Server is running on port 5000
n224387468@cloudshell:~/Deakin-Marketplace$ (sit323-25t1-xiangg-5c638ea)$ kubectl delete pod -l app=deakin-marketplace
pod "deakin-marketplace-948f76c55-9gc6c" deleted
n224387468@cloudshell:~/Deakin-Marketplace$ (sit323-25t1-xiangg-5c638ea)$ kubectl logs -l app=deakin-marketplace
pod "deakin-marketplace-948f76c55-9gc6c" deleted
MongoDB connected successfully
Server is running on port 5000
n224387468@cloudshell:~/Deakin-Marketplace$ (sit323-25t1-xiangg-5c638ea)$ 

```

URL test



Modify the front-end link and use the back-end deployed <http://34.129.60.74>



```

import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import axios from 'axios';
const API_BASE_URL = "http://34.129.60.74";
function CreateListing() {

  const [formData, setFormData] = useState({
    title: '',
    description: '',
    price: '',
    category: ''
  });

  const handleFormChange = (e) => {
    const { name, value } = e.target;
    setFormData({ ...formData, [name]: value });
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    await axios.post(`${API_BASE_URL}/api/items`, formData);
    alert('Item created successfully');
  };
}

```

The front-end and back-end are deployed in the same process

The screenshot shows the Cloud Shell Editor interface with the following details:

- File Explorer:** Shows a project structure with files like `Dockerfile-backend`, `deployment.yaml`, `nginx.conf`, `service.yaml`, `index.js`, and `Dockerfile`.
- Terminal:** The terminal window displays the command `docker build -t shichengxiang/deakin-frontend:v1` being run.
- Output:** The output shows the Docker build process, including layer creation for files like `client`, `node_modules`, and `nginx.conf`, and the final step `[*] Building 5.8s (12/12) FINISHED`.
- Status Bar:** Shows the current file is `Dockerfile-backend`, and other tabs include `deployment.yaml`, `nginx.conf`, `service.yaml`, `index.js`, and `Dockerfile`.
- Bottom Bar:** Includes icons for Cloud Code, Cloud Shell, and Cloud Marketplace, along with a "Hello," message and layout settings.

Log View

Set the GCP project for the current operation

List the Kubernetes clusters created under the current project.

Get cluster credentials and configure kubectl to communicate with the GKE cluster

```
zone: australia-southeast
s224387468@cloudshell:~ (sit323-25t1-xiang-5c638ea)$ Welcome to Cloud Shell! Type "help" to get started.
To set your Cloud Platform project in this session use `gcloud config set project [PROJECT_ID]`.
You can view your projects by running `gcloud projects list`.
s224387468@cloudshell:~ gcloud container clusters list
ERROR: (gcloud.container.clusters.list) The required property [project] is not currently set.
It can be set on a per-command basis by re-running your command with the [--project] flag.

You may set it for your current workspace by running:

$ gcloud config set project VALUE

or it can be set temporarily by the environment variable [CLOUDSDK_CORE_PROJECT]
s224387468@cloudshell:~ gcloud projects list
PROJECT_ID: des-webana-datadictionary
NAME: Des WebAna DataDictionary
PROJECT_NUMBER: 1038693078675

PROJECT_ID: sit323-25t1-xiang-5c638ea
NAME: sit323-25t1-xiang-5c638ea
PROJECT_NUMBER: 986788416464
s224387468@cloudshell:~ gcloud config set project sit323-25t1-xiang-5c638ea
Updated property [core/project].
s224387468@cloudshell:~ (sit323-25t1-xiang-5c638ea)$ gcloud container clusters list
NAME: marketplace-cluster
LOCATION: australia-southeast
MASTER_VERSION: 1.32.3-gke.1785003
MASTER_IP: 34.129.217.139
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.32.3-gke.1785003
NUM_NODES: 3
STATUS: RUNNING
```

View the status of all Pods in the current namespace.

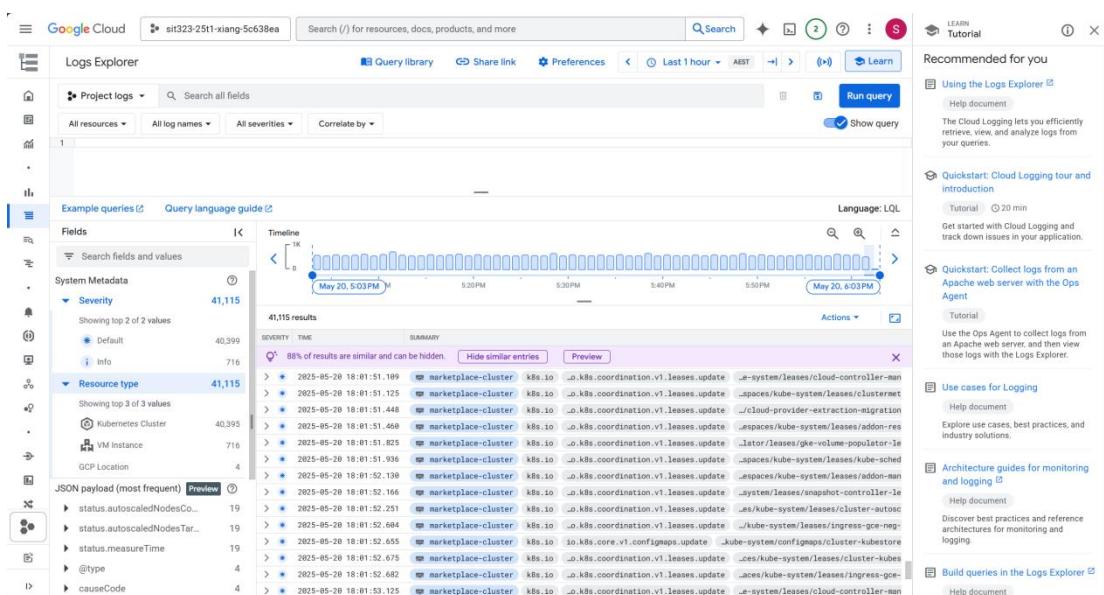
View the details of the Pod, including the IP address and the node where it resides.

Output the Cloud Logging log viewer link and paste it into your browser to access it.

```
s22438746@cloudshell: ~ (sit323-25t-xiang-5c638ea)$ echo "https://console.cloud.google.com/logs/viewer?project=$(gcloud config get-value project)"  
Your active configuration is: [cloudshell-14290]  
https://console.cloud.google.com/logs/viewer?project=sit323-25t-xiang-5c638ea
```

Outputs a link to the Metrics Explorer (graph monitoring) page.

```
sudo messages -j
[sudo] password for sitedash: (sit323-25t1-xiang-5c638ea) $ echo "https://console.cloud.google.com/monitoring/metrics-explorer?project=$(gcloud config get-value project)"
Your active configuration is: [cloudshell-14290]
https://console.cloud.google.com/monitoring/metrics-explorer?project=sit323-25t1-xiang-5c638ea
```



Final results display, all page functions can be used normally, but the pictures cannot be displayed normally

Possible reasons why the image cannot be displayed: the backend does not correctly expose the /uploads static resource.

My container image does not contain the actual uploads folder data because it is statically packaged when building the image. The uploaded files are stored in the container memory and will be lost once it is restarted or redeployed.

 DEAKIN
MARKETPLACE

Home Browse Items Sell Item Messages Profile Logout

Welcome to Deakin Marketplace

Buy and sell second-hand items with fellow Deakin students

[Browse Items](#)



Hot Items



desk
\$70

Likes: 2 [View](#)



Introduction to Programming with Java
Introduction to Programming with Java
\$25

Likes: 2 [View](#)



Dell XPS 13 Laptop (i7, 16GB RAM)
Dell XPS 13 Laptop (i7, 16GB RAM)
\$650

Likes: 2 [View](#)

 DEAKIN
MARKETPLACE

Home Browse Items Sell Item Messages Profile Logout

Filters

Category: All Categories

Condition: Any Condition

Price Range: Min Max

Status: All Status



books
some books Sold

\$20 [View Details](#)



laptop
A two-year-old tablet Pending

\$460 [View Details](#)

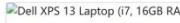


desk
A newly bought desk with two dr... Available

\$70 [View Details](#)



Introduction to Programming with Java



Dell XPS 13 Laptop (i7, 16GB RAM) Available



Study Desk – White IKEA Style Available

 DEAKIN
MARKETPLACE

Home Browse Items Sell Item Messages Profile Logout

Conversations

Introduction to Programming with Java with 123

desk with test

books with ShiCheng Xiang

desk Conversation with test [View Item](#)

aaa 2025/5/19 21:09:32

hi 2025/5/19 21:09:24

hi 2025/5/18 22:25:45

aaa 2025/5/18 22:25:12

ok 2025/5/18 22:20:48

Type your message... [Send](#)

 **Profile**

Name: ShiCheng Xiang
Email: s224387468@deakin.edu.au
Phone: 1111

[Edit Profile](#)

My Listings

[Sell New Item](#)

Category	Description	Price	Status	Action
books	Introduction to Programming with Java	\$25	Available	View Details
laptop	Introduction to Programming with Java	\$460	Pending	Mark as Sold Back to Available View Details
desk	Introduction to Programming with Java	\$70	Sold	Transaction Completed View Details

 [Home](#) [Browse Items](#) [Sell Item](#) [Messages](#) [Profile](#) [Logout](#)

[Back to Items](#)

 Introduction to Programming with Java

\$25

Category: textbooks
Condition: fair
Location: Burwood Campus Library, Level 2
Description: Good condition, includes notes and highlights from previous semester.
Posted by: ShiCheng Xiang

Liked (2)

[Contact Seller](#)

Sell an Item

Title

Description

Price (\$)

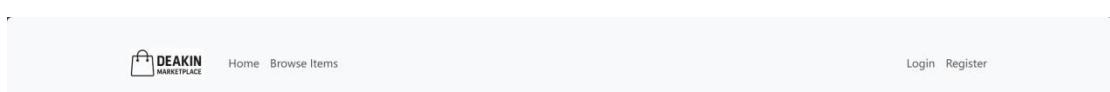
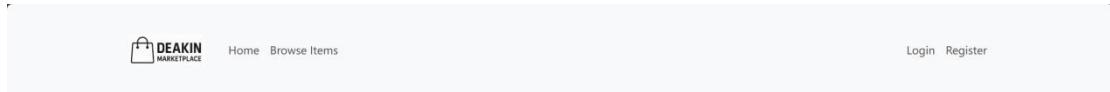
Category

Condition

Image
 No file chosen

Location

[Create Listing](#)



Normal display effect of the picture

Welcome to Deakin Marketplace

Buy and sell second-hand items with fellow Deakin students

[Browse Items](#)

Hot Items

desk
\$70
Likes: 2
[View](#)

Introduction to Programming with Java
\$25
Likes: 2
[View](#)

Dell XPS 13 Laptop (i7, 16GB RAM)
\$650
Likes: 2
[View](#)

[DEAKIN MARKETPLACE](#) Home Browse Items Sell Item Messages Profile [Logout](#)

Filters

Category: All Categories

Condition: Any Condition

Price Range: Min Max

Status: All Status

books
some books
\$20 [View Details](#)

laptop
A two-year-old tablet
\$460 [View Details](#)

desk
A newly bought desk with two dr...
\$70 [View Details](#)

Java™ Programming and Data Structures
Comprehensive Version
Y. Daniel Liang
TWELFTH EDITION

Introduction to Programming with

Dell XPS 13 Laptop (i7, 16GB RAM) [Available](#)

Study Desk – White IKEA Style [Available](#)

[DEAKIN MARKETPLACE](#) Home Browse Items Sell Item Messages Profile [Logout](#)

Profile

Name: ShiCheng Xiang
Email: s224387468@deakin.edu.au
Phone: 1111
[Edit Profile](#)

My Listings

[Sell New Item](#)

books
Price: \$20
Status: Sold
Transaction Completed
[View Details](#)

laptop
Price: \$460
Status: Pending
Mark as Sold [Back to Available](#)
[View Details](#)

and Data Structures
Comprehensive Version
Y. Daniel Liang
TWELFTH EDITION

README.md file updated

The screenshot shows a README.md file with the following content:

Build & Deploy Workflow

1. Git Configuration (Cloud Shell)

```
git config --global user.name "xsc224387468"  
git config --global user.email "you@example.com"
```

2. Build & Push Docker Images

Backend

```
docker build -t shichengxiang/deakin-marketplace:v5 -f Dockerfile-backend .  
docker push shichengxiang/deakin-marketplace:v5
```

Frontend

```
cd client  
npm install  
npm run build  
cd ..  
docker build -t shichengxiang/deakin-frontend:v9 -f Dockerfile-frontend .  
docker push shichengxiang/deakin-frontend:v9
```

3. Apply Kubernetes Configuration

Backend

```
kubectl apply -f deployment.yaml  
kubectl apply -f service.yaml
```

Frontend

```
kubectl apply -f client/deployment-frontend.yaml  
kubectl apply -f client/service-frontend.yaml
```

The screenshot shows a README.md file with the following content:

4. Check LoadBalancer IPs

```
kubectl get service
```

Example:

- Frontend: <http://34.129.26.216>
- Backend API: <http://34.129.66.74/api>

Image Upload & Persistent Storage

- Uploaded images are stored in the `/uploads` directory
- Kubernetes mounts this directory using `emptyDir` or a `PersistentVolumeClaim`
- The backend exposes it via:

```
app.use('/uploads', express.static(path.join(__dirname, '../uploads')));
```

Functional Pages

The screenshot shows a GitHub README page for a project named "Deakin Marketplace". The page includes sections for "Functional Pages", "Security Notes", and "Push to GitHub from Cloud Shell".

- Functional Pages:**
 - /register & /login : JWT-based auth
 - /sell : Form with image upload
 - /items : Public marketplace
 - /profile : Lists user's items and liked items
 - /messages : Chat interface
- Security Notes:**
 - MongoDB URI is loaded via `process.env.MONGODB_URI`
 - Frontend API endpoint is hardcoded:

```
const API_BASE_URL = "http://34.129.60.74";
```
- Push to GitHub from Cloud Shell:**

```
git init
git remote add origin https://github.com/xsc224387468/Deakin-Marketplace.git
git add .
git commit -m "Initial commit from Cloud Shell"
git push -u origin main
```
- Author:**
 - Shicheng Xiang
 - Student ID: s224387468
 - GitHub: [xsc224387468](https://github.com/xsc224387468)

Visit the website

[Deakin Marketplace](https://github.com/xsc224387468/Deakin-Marketplace)

PPT



Function Introduction

- Product list display
- Product details
- Upload Products (Including image upload)
- Like/Unlike function
- Send Message
- Login/Register
- Product status management (available, pending, sold)
- Search and filter
- Modify personal information

Problems

- Permission issues
- Subnet/VPC network configuration issues
- Failed to create LoadBalancer service access (e.g. curl time out)
- Front-end page navigation path reports 404
- Environment variables are not effective
- Failed to upload image
- The image cannot be loaded (e.g. /uploads/avatar-xxxx.jpg)

Summary

- Project Architecture

- MongoDB database design(External hosting, no containerized deployment required)
- React + Nginx to build the frontend(deakin-frontend Deployment + deakin-frontend-service)
- Node.js + MongoDB Atlas deployment backend(deakin-marketplace Deployment + deakin-marketplace-service)

- Microservice architecture

- Separate deployment of front-end and back-end
- Use independent services to handle different functions (product services, message services, user services)

- Containerized deployment

- Use Docker containerized applications
- Easy to deploy and scale on GCP
- Improve application portability



Week	Work content	Problems and Solutions
Week 4	Submit 4.3HD plan, design system architecture and components	
Week 5	Build Node.js + Express backend, design API structure	
Week 6	Build front-end React project, integrate Bootstrap and React Router	Initial page cannot jump, fix Router configuration
Week 7	Configure MongoDB Atlas, complete basic CRUD functions	MongoDB URI is not configured correctly, fix it through unified management of .env file
Week 8	Docker containerize front-end and back-end, write Dockerfile and docker-compose	Static resources (such as pictures) are not configured correctly in nginx, fix build path
Week 9	Kubernetes deployment: build cluster, configure Deployment and Service	Insufficient permissions and cluster configuration failure occurred, switched projects, enabled subnets and configured firewalls to solve
Week 10	Configure Stackdriver Monitoring and Logging	No permission to access the monitoring panel in the early stage, solved by correctly switching projects and redeploying

		services
Week 11	Shoot demonstration video, prepare demonstration PPT, collect screenshots and GitHub submissions	Some pages are inaccessible, fixed by separating the front and back ends and explicitly exposing ports of independent services

Project Description

1. Number of clusters and nodes

- A total of 1 Kubernetes cluster was created (in the GCP GKE console), and services were deployed multiple times using kubectl apply.
- By default, each cluster contains 1 node pool and 3 nodes (not manually scaled down, so it is the default configuration).
- The default-pool contains 3 e2-medium type nodes by default (each with 2vCPU + 4GB RAM).

2. Permission issues encountered in the early stage and solutions

Problem: When trying to make some configurations on GCP (such as deploying or enabling services), an error is reported:

Cause:

When using the GCP account provided by the school Deakin, the account has insufficient default permissions and cannot fully control the project or resources (for example, the Kubernetes Engine API cannot be enabled, service accounts cannot be created, etc.).

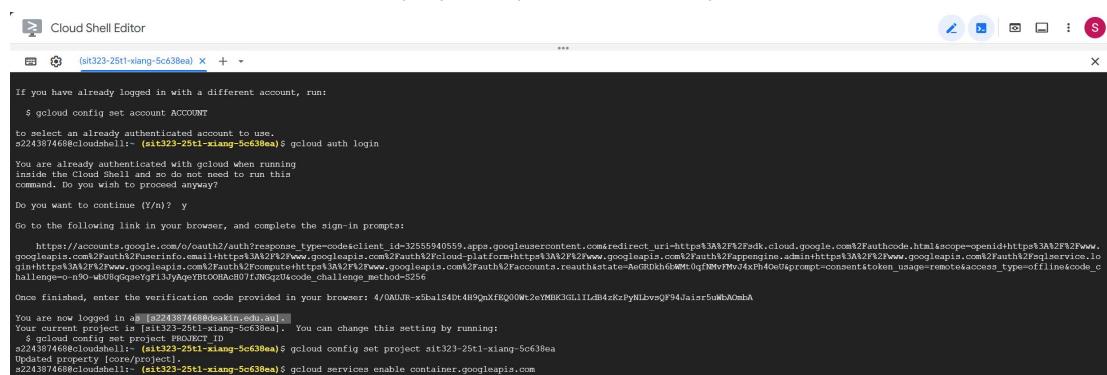
GCP requires you to have the Editor or Owner role of the project to operate resources.

Solution:

Abandon using the restricted school account, directly use a personal Google account (with full permissions) to recreate the GCP project, and configure permissions from scratch.

Use gcloud init to initialize the GCP CLI and configure a new project.

Use Cloud Shell to continue the deployment process to avoid permission barriers.



```

Cloud Shell Editor
(sit323-25t1-xiang-5c638ea) ✘ + ⌂ ⓘ

If you have already logged in with a different account, run:
$ gcloud config set account ACCOUNT
to select an already authenticated account to use.
s22438746@cloudshell:~ (sit323-25t1-xiang-5c638ea)$ gcloud auth login

You are already authenticated with gcloud when running
inside the Cloud Shell and so do not need to run this
command. Do you wish to proceed anyway?

Do you want to continue (Y/n)? y

Go to the following link in your browser, and complete the sign-in prompts:
https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=3255940559.apps.googleusercontent.com&redirect_uri=https%3A%2F%2Fcloud.google.com%2Fauth%2Fcode&scope=https://www.googleapis.com%2Fauth%2Ftoken&admin=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts%3AreadState=AgBpkGKwM%0fNm%74xPh4oU&prompt=consent&token_usage=remote&access_type=offline&code_challenge_method=S256

Once finished, enter the verification code provided in your browser: 4/0AUJR-x5ba1S4Dt4H9QnxFBQ00Wt2eYMBK3GL11LdBi4zKzPyNLkvnQF94Jaisr5uWbK0mb

You are now logged in as [s22438746@cloudshell:~].
Your current project is [sit323-25t1-xiang-5c638ea]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
s22438746@cloudshell:~ (sit323-25t1-xiang-5c638ea)$ gcloud config set project sit323-25t1-xiang-5c638ea
Updated property [core/project].
s22438746@cloudshell:~ (sit323-25t1-xiang-5c638ea)$ gcloud services enable container.googleapis.com

```

```

Cloud Shell Editor
Normal EnsureLoadBalancer 4m36s service-controller Ensured load balancer
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ grep image deployment.yaml
    image: australia-southeast2-docker.pkg.dev/sit323-25tl-xiang-5c638ea/marketplace-repo/deakin-marketplace:v1
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ gcloud projects add-iam-policy-binding sit323-25tl-xiang-5c638ea \
--role=roles/artifactregistry.reader
ERROR: (gcloud.projects.addIamPolicyBinding) [s224387468@deakin.edu.au] does not have permission to access projects instance [sit323-25tl-xiang-5c638ea:metiamPolicy] (or it may not exist): Policy update access denied. User is authenticated as s224387468@deakin.edu.au which is the active account specified by the [core/account] property.
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ gcloud artifacts repositories add-iam-policy-binding marketplace-repo \
--location=australia-southeast2 \
--members=artifactregistry.reader
--role=roles/artifactregistry.read
ERROR: (gcloud.artifacts.repositories.addIamPolicyBinding) PERMISSION DENIED: Permission 'artifactregistry.repositories.setIamPolicy' denied on resource '//artifactregistry.googleapis.com/projects/sit323-25tl-xiang-5c638ea/locations/australia-southeast2/repositories/marketplace-repo' (or it may not exist). This command is authenticated as s224387468@deakin.edu.au which is the active account specified by the [core/account] property.
- *type*: https://www.googleapis.com/google.rpc.ErrorInfo
  domain: artifactregistry.googleapis.com
metadata:
  permission: artifactregistry.repositories.setIamPolicy
  resource: projects/sit323-25tl-xiang-5c638ea/locations/australia-southeast2/repositories/marketplace-repo
  reason: IAM_PERMISSION_DENIED
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker tag australia-southeast2-docker.pkg.dev/sit323-25tl-xiang-5c638ea/marketplace-repo:shichengxiang/deakin-marketplace:v1
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker login
USING WEB-BASED LOGIN

Info - To sign in with credentials on the command line, use 'docker login -u <username>'.

Your one-time device confirmation code is: DKCL-TDFX
Press ENTER to open your browser or submit your device code here: https://login.docker.com/activate
Waiting for authentication in the browser...
~Clogin canceled
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker login -u shichengxiang
Info - A Personal Access Token (PAT) can be used instead.
To create a PAT, visit https://hub.docker.com/secrets

Password:
WARNING! Your credentials are stored unencrypted in '/home/s224387468/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/
Login Successful
s224387468@cloudshell:/~Deakin-Marketplace (sit323-25tl-xiang-5c638ea)$ docker tag australia-southeast2-docker.pkg.dev/sit323-25tl-xiang-5c638ea/marketplace-repo:deakin-marketplace:v1 xcdockeruser/deakin-mark

```

3. Failed to create access to LoadBalancer service (connection channel problem when exposing services to the public network)

Problem: Access to the front-end or back-end service failed after being exposed through the LoadBalancer type (such as curl timeout).

Cause:

Container listening address is not 0.0.0.0

Service targetPort is inconsistent

Firewall rules are not enabled

Solution:

Use app.listen(5000, '0.0.0.0') in the backend server/index.js

Service configuration targetPort: 5000

Open GCP network firewall rules to allow TCP ports 80 and 5000 to enter

<input type="checkbox"/>	Name	Type	Targets	Filters	Protocols / ports	Action	
<input type="checkbox"/>	gke-marketplace-cluster-30d84921-inkubelet	Ingress	gke-	IP ranges:	tcp:10255	Allow	
<input type="checkbox"/>	gke-marketplace-cluster-30d84921-exkubelet	Ingress	gke-	IP ranges:	tcp:10255	Deny	
<input type="checkbox"/>	allow-http-external	Ingress	Apply to all	IP ranges:	tcp:80, 5000	Allow	
<input type="checkbox"/>	gke-marketplace-cluster-30d84921-all	Ingress	gke-	IP ranges:	tcp udp sctp icmp;esp;ah	Allow	
<input type="checkbox"/>	gke-marketplace-cluster-30d84921-vms	Ingress	gke-	IP ranges:	tcp:1-65535 udp:1-65535 icmp	Allow	
<input type="checkbox"/>	k8s-13f48208d4b675f9-node-http-hc	Ingress	gke-	IP ranges:	tcp:10256	Allow	
<input type="checkbox"/>	k8s-fw-a6e149f394a5945898564db42cffc3fd	Ingress	gke-	IP ranges:	tcp:80	Allow	
<input type="checkbox"/>	k8s-fw-a866965e711e34bb8836195abdbf3020	Ingress	gke-	IP ranges:	tcp:5000	Allow	
<input type="checkbox"/>	k8s-fw-aea7b7c40f3454eaca4a05b1a884db5d	Ingress	gke-	IP ranges:	tcp:80	Allow	

4. Front-end page navigation path reports 404

Problem: For example, 404 is reported when accessing the /profile page

Cause: When React Router uses BrowserRouter mode, non-root paths are not processed by index.html, and Nginx cannot find static resources directly

Solution:

Add Nginx routing fallback rules in client/Dockerfile-frontend:



```
nginx.conf
server {
    listen 80;
    server_name localhost;

    root /usr/share/nginx/html;
    index index.html;

    location / {
        try_files $uri /index.html;
    }

    location /static/ {
        expires 1y;
        access_log off;
    }
}
```

5. Environment variables are not effective

Problem: REACT_APP_API_URL in .env.production is invalid

Cause: React environment variables need to be in Write when building, so .env.production must exist in the build context before npm run build

Solution:

Write the backend URL in each page file

```
const API_BASE_URL = "http://34.129.60.74";

const response = await axios.get(`${API_BASE_URL}/api/items?seller=${currentUser.id}` , {
  headers: {
    'Authorization': `Bearer ${localStorage.getItem('token')}`
  }
})
```

6. Failed to upload image

Problem: Error when uploading products, no image received

Reason:

Multer set path error or FormData structure does not match

Solution:

Multer configuration:

```
const storage = multer.diskStorage({
  destination: (req, file, cb) => cb(null, 'uploads/'),
  filename: (req, file, cb) => cb(null, `image-${Date.now()}-${path.extname(file.originalname)})`);
```

Add Deployment volumeMounts and Service port mapping

Use formData.append('images', file) to upload file fields consistent

7. Image cannot be loaded (such as /uploads/avatar-xxxx.jpg) (not yet resolved)

Problem: The browser console shows 404, the path is similar to /uploads/xxx.jpg

Possible cause: The uploads/ folder in the container is not mounted correctly or not exposed as a static resource (Not mounted correctly or not exposed as a static resource)