



Avenue des Facultés
80000 Amiens

Documentation Docker

SAE Semestre 5

Liam TARGET, Justin LARGILLIÈRE, Malcolm WALTER, Lucas GONTHIER, Lorenzo VATRIN, Victor DA SILVA LOURENCO MARQUES.

BUT3 Informatique
Tous parcours

Année

2024 - 2025



Documentation Docker v3.2

Mise en place des fichiers Docker et modification des ports

Docker gère des conteneurs individuels, tandis que Docker Compose orchestre plusieurs conteneurs.

Au préalable :

- Docker Desktop allumé

En ligne de commande dans le dossier de travail :

- Ligne de commande, dans le dossier de travail : `docker compose up --build -d`
- Pour éteindre le projet : `docker compose down`

Bonnes pratiques :

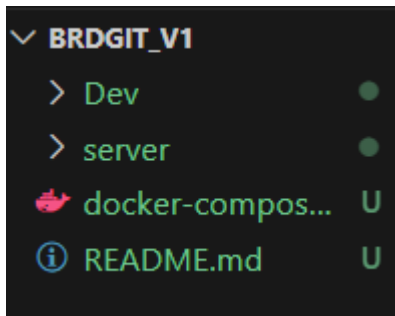
- Avant de construire une image on peut vérifier les dépendances avec `npm audit`

Vérifications :

```
PS C:\Users\liamt\Documents\SAE_S5_BRDgame\Dev> npm audit
found 0 vulnerabilities
```

```
PS C:\Users\liamt\Documents\SAE_S5_BRDgame\server> npm audit
found 0 vulnerabilities
```

Emplacement du docker-compose.yml : Racine du projet



Dans le dossier server :

- Le Dockerfile pour le server (backend)



BRDGame

```
server > Dockerfile > ...
1  # Utiliser l'image de Node.js comme base
2  FROM node:18-alpine
3
4  # Créer un répertoire de travail
5  WORKDIR /app
6
7  # COPY du package.json pour vérifier les dépendances et éviter les versions non-testées
8  COPY package*.json .
9
10 # Installation des dépendances
11 RUN npm install
12
13 # Copier les fichiers de l'application dans le conteneur
14 COPY . .
15
16 # Exposer le port
17 EXPOSE 3000
18
19 #RUN npm init -y Pour crée un package.json
20
21 # Installation du client socket.io
22 RUN npm install express socket.io
23 RUN npm install socket.io-client
24
25 # Commande pour démarrer le serveur
26 CMD ["node", "server.js"]
27
28 #Dockerfile server
```

Dans la configuration du server :

- Configurer les ports pour les ouvrir au frontend

```
server > JS server.js > ioServer.on('connection') callback
26  const ioServer = new socketIO.Server(server, {
31  |  }
32  });
33
34  const PORT = process.env.PORT || 3000;
35  app.listen(PORT, () => {
36  |  console.log(`Server is running on port ${PORT}.`);
37  |  });
38
39  createInitialFarmables()
40
41  ioServer.on('connection', (socket) => {
42  |
43  |  players.push({ x: 0, y: 0, id: socket.id, inGame: false, hp: 0});
44  |
45  |  console.log(`A player connected: ${socket.id}`);
46  |  });
```

Dans le dossier Dev :

- Le Dockerfile pour l'application (frontend)



The screenshot shows the VS Code interface with the Explorer on the left and the Dockerfile editor in the center. The Explorer shows the project structure for SAE_S5_BRDGAME, including a Dev folder with assets, Classes, Modules, Scenes, .gitignore, Dockerfile, index.html, main.js, package-lock.json, package.json, style.css, vite.config.js, and a server folder. The Dockerfile editor shows the following content:

```

1  # Utiliser l'image de Node.js comme base
2  FROM node:18-alpine
3
4  # Créer un répertoire de travail
5  WORKDIR /app
6
7  # COPY du package.json
8  COPY package*.json .
9
10 # Installation des dépendances
11 RUN npm install
12
13 # Copier les fichiers de l'application dans le conteneur
14 COPY . .
15
16 # Exposer le port
17 EXPOSE 8000
18
19 # Commande pour démarrer le serveur
20 CMD ["npm", "run", "dev", "--", "--host"]
21
22 #Dockerfile Dev
  
```

La configuration du port de Vite :

The screenshot shows the VS Code interface with the Explorer on the left and the vite.config.js editor in the center. The Explorer shows the project structure for SAE_S5_... with a Dev folder containing assets, Classes, Modules, Scenes, .gitignore, Dockerfile, index.html, main.js, package-lock.json, package.json, style.css, vite.config.js, and a server folder. The vite.config.js editor shows the following content:

```

1  import { defineConfig } from 'vite';
2
3  export default defineConfig({
4    server: {
5      port: 8000,
6    },
7  });
  
```

Pour la configuration du Docker compose :



```

SAE_S5_...  docker-compose.yml
  > Dev
  > server
  docker-compose.yml
  README.md

1  services:
2
3  phaser-app: # frontend
4    build: ./Dev/
5    ports:
6      - "8000:8000"
7    restart: always
8    networks: # Permet de communiquer uniquement au backend
9      - server
10   mem_limit: 512m # Limite de mémoire
11   cpus: "0.5" # Limite de CPU (50% d'un core)
12
13  server: # backend
14    build: ./server/
15    ports:
16      - "3000:3000"
17    restart: always
18    networks: # Permet de communiquer avec le frontend et la bdd
19      - Dev
20   mem_limit: 1g # Limite de mémoire
21   cpus: "1.0" # Limite de CPU (1 core)
22   logging:
23     driver: "json-file" # Spécifie le driver de log à utiliser
24     options:
25       max-size: "10m" # Limite la taille du fichier de log
26       max-file: "3" # Limite le nombre de fichiers de log
27
28
29
30  networks:
31    Dev:
32      driver: bridge # Réseau de communication privé entre pls conteneurs
33    server:
34      driver: bridge # Réseau de communication privé entre pls conteneurs

```

Commandes à utiliser : Avec docker et docker compose

Pour afficher les conteneurs actif : `docker ps`

Pour afficher les logs : `docker logs`

Pour afficher les réseau : `docker network ls`

Pour Lancer les services : `docker compose up` (option -d pour démon en arrière-plan)

Pour une modification à chaud et recréer les conteneurs : `docker compose up --build`

Pour arrêter les service `docker compose down`

Un docker stats pour vérifier la gestion de consommation de ressources et optimiser les performances:

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
a2cd10c8521e	sae_s5_brdgame-server-1	0.00%	17.46MiB / 7.441GiB	0.23%	1.57kB / 0B	0B / 0B	7
cb477df7ed7f	sae_s5_brdgame-phaser-app-1	0.08%	253.6MiB / 7.441GiB	3.33%	1.57kB / 0B	0B / 0B	36

Nous sommes en Serveur dédié normalement avec les serveurs de l'iut :



Cela implique un coût élevé, plus de complexité à mettre en place en revanche très fiable sinon cela dépendra du serveur lui-même c'est adapté pour un nombre important de joueurs. C'est le meilleur en termes de sécurité.