

Challenge 3 – Docker Deployment

Projects:

- RoyalChess (MERN stack)
- STAC Website (Next.js + Django)

Objective:

Deploy both projects using Docker containers with Docker Compose, configure internal networking, and ensure databases are not exposed outside containers.

Objective

- Containerize both projects.
- Run both stacks simultaneously using Docker Compose.
- Configure internal Docker network so databases are inaccessible from host.
- Document full deployment process and issues faced.

Initial Setup:

I cloned both repositories inside:

C:\Users\devan\OneDrive\Scans\saic\challenge3

Folder structure:

```
challenge3
├── royalchess
├── stac-clone
└── docker-compose.yml
```

Name	Status	Date modified	Type	Size
royalchess	⟳	18-01-2026 10:57	File folder	
stac-clone	⟳	18-01-2026 12:04	File folder	
docker-compose.yml	✗	18-01-2026 14:39	Yaml Source File	2 KB

Steps Performed

Step 1 — Repository Inspection

•RoyalChess

- client/ → React frontend
- server/ → Node backend with TypeScript & WebSockets

•STAC Clone

- frontend/ → Next.js frontend
- backend/ → Django backend

Step 2 — Dockerfile Creation

Dockerfiles were created for:

- RoyalChess frontend
- RoyalChess backend
- STAC frontend
- STAC backend

Each Dockerfile performed:

- Copy dependency files
- Install dependencies
- Build project
- Expose required ports

Step 3 — Docker Compose Setup

A single `docker-compose.yml` was created defining:

Services

- royal_db (MongoDB)
- royal_backend (Node server)
- royal_frontend (React client)
- stac_db (PostgreSQL)
- stac_backend (Django server)
- stac_frontend (Next.js client)

Network

- Custom bridge network: challenge3 internal net

Security Implementation:

- Databases had no host-exposed ports.
- Only backend services connected to databases internally.
- Ensured databases inaccessible from host or outside environment.

Issues Faced & Fix Attempts

✗ Issue 1 — VS Code & Path Errors

- Folder rename failed.
- VS Code path broke temporarily.

Fix:

Reinstalled VS Code and reopened workspace.

```
[T] up 27/27] devan [challenge3] docker compose up -d --build
✓ Image mongo:6      Pulled                               77.1s
✓ Image postgres:15   Pulled                               58.1s
- Image challenge3-royal_frontend Building               1.4s
- Image challenge3-royal_backend Building                1.4s
- Image challenge3-stac_frontend Building               1.4s
- Image challenge3-stac_backend Building                1.4s
unable to prepare context: path "C:\\\\Users\\\\devan\\\\OneDrive\\\\Scans\\\\saic\\\\challenge3\\\\stac\\\\backend" not found

::\Users\devan\OneDrive\Scans\saic\challenge3>docker compose up -d --build
time="2026-01-18T12:08:38+05:30" level=warning msg="C:\\\\Users\\\\devan\\\\OneDrive\\\\Scans\\\\saic\\\\challenge3\\\\docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion"
unable to get image 'postgres:15': error during connect: Get "http://%2F%2Fpipe%2FdockerDesktopLinuxEngine/v1.51/images/postgres:15/json": open //./pipe/dockerDesktopLinuxEngine: The system cannot find the file specified.

::\Users\devan\OneDrive\Scans\saic\challenge3>
```

✗ Issue 2 — Docker Build Context Path Errors

unable to prepare context: path not found

Fix:

Corrected folder names and directory paths.

Issue 3 — RoyalChess Node Engine Conflict

Unsupported engine: wanted node >=20
current node v18.20.8

Fix Attempted:

- Updated Docker image to Node 20.
- Retried pnpm installs.

Result:

Dependencies still failed; tsc-watch and next not found inside container.

Issue 4 — RoyalChess pnpm Workspace Errors

Local package.json exists but node_modules missing

pnpm run build failed

Fix Attempted:

- Installed pnpm globally inside container.
- Adjusted workspace build commands.

Result:

Build still failed repeatedly.

Issue 5 — STAC Frontend Missing Dependency

Module not found: next-themes

Fix:

Installed missing dependency and rebuilt container.

Result: STAC frontend built successfully.

Issue 6 — STAC Backend Missing Module

ModuleNotFoundError: No module named 'corsheaders'

Fix:

Added django-cors-headers to requirements.txt.

Rebuilt container.

Result: Django backend started successfully.

Issue 7 — STAC Frontend “Failed to load homepage data”

Cause:

Frontend initially could not communicate with backend API.

Fix:

Adjusted API base URL to use internal Docker network hostname.

Rebuilt containers.

Result:

STAC website loaded properly.

✖ Issue 8 — RoyalChess Containers Breaking Compose

Stack

RoyalChess backend build failures repeatedly crashed compose process and sometimes removed working STAC containers.

Fix:

Deployed only STAC stack separately:

```
docker compose up -d --build stac_db
```

```
stac_backend stac_frontend
```

STAC stack recovered and worked.

Final Container State

At the end, Docker successfully built **all six containers**:

```
challenge3-royal_db-1
```

```
challenge3-royal_backend-1
```

```
challenge3-royal_frontend-1
```

```
challenge3-stac_db-1
```

```
challenge3-stac_backend-1
```

```
challenge3-stac_frontend-1
```

⚠ Final Functional Status					
Project	Frontend	Backend	Database	Status	
STAC Website	✓ Working	⚠ Partially Working	✓ Working	Site loads but homepage API still unstable	
RoyalChess	⚠ UI Loads	✗ Backend Not Working	✓ DB Working	Login/game flow blocked	

What Could Not Be Fully Achieved

RoyalChess Backend

- Persistent Node ≥20 engine mismatch.
- pnpm workspace build failures.
- tsc-watch missing at runtime.
- Backend container started but never served functional API.

STAC Backend

- Container ran but homepage data API occasionally failed even after multiple code fixes.

Despite trying multiple Dockerfile and dependency fixes, both backends did not reach fully stable production-ready behavior.

Security Requirements Achieved

- ✓ Databases not exposed to host
 - ✓ Only backend containers have DB access
 - ✓ Internal Docker bridge network used
 - ✓ No external unauthorized DB access possible
- Security objective satisfied.

🏁 Conclusion

- Docker Compose orchestration achieved successfully
- Internal network and database isolation implemented
- STAC frontend and backend containers deployed
- RoyalChess UI deployed

However:

Both backend services could not be fully stabilized even after multiple dependency and configuration fixes. Still, containerization, networking, and security objectives were implemented correctly

```

> [royal_frontend 4/7] RUN npm install --production:
0.908 npm warn config production Use '--omit=dev' instead.
17.32 npm error code EUNSUPPORTEDPROTOCOL
17.32 npm error Unsupported URL Type "workspace": workspace:*
17.33 npm notice
17.33 npm notice New major version of npm available! 10.8.2 -> 11.7.0
17.33 npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.7.0
17.33 npm notice To update run: npm install -g npm@11.7.0
17.33 npm notice
17.34 npm error A complete log of this run can be found in: /root/.npm/_logs
/2026-01-18T06_53_49_414Z-debug-0.log
-----
[+] up 0/4
- Image challenge3-stac_backend Building      53.6s
- Image challenge3-royal_frontend Building     53.6s
- Image challenge3-royal_backend Building      53.6s
- Image challenge3-stac_frontend Building     53.6s
Dockerfile:4
-----
2 |   WORKDIR /app
3 |   COPY package*.json .
4 | >>> RUN npm install --production
5 |   COPY . .
6 |   RUN npm run build
-----
target royal_frontend: failed to solve: process "/bin/sh -c npm install --pr
oduction" did not complete successfully: exit code: 1
target royal_backend: failed to solve: failed to compute cache key: failed t
o calculate checksum of ref v6915wr25z4ty8tbxufjt09d0::bc98g76qf9mlga536ag76
58a2: "/pnpm-lock.yaml": not found

```

C:\Users\devan\OneDrive\Scans\saic\challenge3>docker compose up -d --build
services must be a mapping

C:\Users\devan\OneDrive\Scans\saic\challenge3>

```

C:\Users\devan\OneDrive\Scans\saic\challenge3>
C:\Users\devan\OneDrive\Scans\saic\challenge3>docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
9e7c7e25c0be challenge3-stac_frontend "docker-entrypoint.s_" 3 minutes ago Up 3 minutes 0.0.0.0:3002->3000/tcp, [::]:3002->3000/tcp
7948278a975a challenge3-stac_backend "python manage.py ru_" 3 minutes ago Up 3 minutes 8000/tcp
ae7724745f05 postgres:15 "docker-entrypoint.s_" 3 minutes ago Up 3 minutes 5432/tcp
7cfdb0016e0e2 mongo:6 challenge3-royal_db-1 "docker-entrypoint.s_" 3 minutes ago Up 3 minutes 27017/tcp

```

C:\Users\devan\OneDrive\Scans\saic\challenge3>

```

C:\Users\devan\OneDrive\Scans\saic\challenge3>docker compose config --services
royal_db
royal_backend
royal_frontend
stac_db
stac_backend
stac_frontend

```

```

1a104ce0bb2c59e2d4d+88bb02/59+a/045da2c98884/ac39+4ea35de814
-----
> [royal_backend 4/7] COPY package.json pnpm-lock.yaml pnpm-workspace.yaml .
-----
> [royal_frontend 4/8] COPY package.json pnpm-lock.yaml pnpm-workspace.yaml .
-----
[+] up 0/4
- Image challenge3-stac_backend Building      7.2s
- Image challenge3-royal_frontend Building     7.2s
- Image challenge3-royal_backend Building      7.2s
- Image challenge3-stac_frontend Building     7.2s
Dockerfile:7
-----
5 |   RUN npm install -g pnpm
6 |
7 | >>> COPY package.json pnpm-lock.yaml pnpm-workspace.yaml .
8 |
9 |   RUN pnpm install --frozen-lockfile
-----
```

```

#38 unpacking to docker.io/library/challenge3-stac_backend:latest 51.2s done
#38 DONE 169.7s
#39 [stac_backend] resolving provenance for metadata file
#39 DONE 0.1s
[+] up 13/13
✓ Image challenge3-royal_frontend Built
✓ Image challenge3-royal_backend Built
✓ Image challenge3-stac_frontend Built
✓ Image challenge3-stac_backend Built
✓ Network challenge3_internal_net Created
✓ Volume challenge3_royal_data Created
✓ Volume challenge3_stac_data Created
✓ Container challenge3_stac_db-1 Created
✓ Container challenge3_royal_db-1 Created
✓ Container challenge3-stac_backend-1 Created
✓ Container challenge3-royal_backend-1 Created
✓ Container challenge3-stac_frontend-1 Created
✓ Container challenge3-royal_frontend-1 Created

```

```

C:\Users\devan\OneDrive\Scans\saic\challenge3>
C:\Users\devan\OneDrive\Scans\saic\challenge3>
C:\Users\devan\OneDrive\Scans\saic\challenge3>
```

Failed to compile

```

./app/theme-provider.tsx:5:0
Module not found: Can't resolve 'next-themes'
  3 |
  4 | import React, { useEffect, useState } from "react";
> 5 | import { ThemeProvider as NextThemeProvider } from "next-themes";
  6 | // We will let TypeScript infer the props type or use
React.ComponentProps for now
  7 |
  8 | type NextThemeProviderProps = React.ComponentProps<typeof
NextThemeProvider> & {
```

<https://nextjs.org/docs/messages/module-not-found>

This error occurred during the build process and can only be dismissed by fixing the error.