

PIES Studio - Offline Deployment Guide

This guide walks you through setting up the full PIES Studio platform in offline mode using Docker Compose. It is designed for enterprise customers who have received access to licensed containers and configuration files.

Table of Contents

- [Table of Contents](#)
 - [1. Prerequisites](#)
 - [2. Directory Structure](#)
 - [3. Configuration Files](#)
 - [4. Getting Started](#)
 - [Step 1: Install Docker and Docker Compose](#)
 - [Step 2: Prepare Required Files](#)
 - [Step 3: Start the System](#)
 - [5. Docker Compose Setup](#)
 - [PIES Studio Images](#)
 - [Start and Stop Scripts](#)
 - [6. Post-Setup Verification](#)
 - [7. Running on Other Platforms \(Kubernetes, Nomad, etc.\)](#)
 - [8. Individual Installation](#)
 - [9. Support](#)
-

1. Prerequisites

Before you begin, ensure the following are installed on your system:

- Docker (v20+ recommended)
- Docker Compose (v1.29+ or Docker Compose Plugin for v2)
- Linux/Unix environment (Ubuntu/RHEL recommended)
- A valid offline license file `pies_studio.license`

If you're using a VM or air-gapped machine, ensure Docker can access the required ports.

2. Directory Structure

Create a directory structure as follows:


```
1 /pies-studio/
2 |-- docker-compose.yml
3 |-- start.sh
4 |-- stop.sh
5 |-- config/
6 |   |-- license/
7 |   |   |-- config.json
8 |   |   |-- config.web.json
9 |   |   |-- pies_studio.license
10 |   |-- core/config.json
```

```
11 | |-- web/config.json
12 | |-- preview/config.json
13 | |-- codegen/config.json
14 | |-- vault/config/init-response.json
15 |-- certs/
16 | |-- client/
17 |-- db/
18 | |-- mongo/
19 |-- generated/
```

3. Configuration Files

Each microservice container reads its environment-specific configuration from mounted JSON files.

File Path	Purpose
config/license/config.json	Configuration for the License Server
config/core/config.json	Configuration for the PIES Studio Core backend
config/web/config.json	Runtime config for the Studio frontend
config/preview/config.json	Runtime config for the Preview frontend
config/codegen/config.json	Runtime config for the code generation engine
vault/config/init-response.json	Unsealing data for Vault, used by Codegen. Automatically generated, should not be written manually.
pies_studio.license	Your encrypted license key (do not modify)

 Detailed information regarding each configuration file can be found on the 'Repository Overview' section for the image on DockerHub.

The links to the DockerHub repositories can be found in the sections that follow.

For platform-specific deployments (Kubernetes, Nomad, etc.), ensure these configuration files are:

- Mounted into containers using ConfigMaps or host volumes
- Set to read-only where applicable
- Maintained securely, especially pies_studio.license

4. Getting Started

Step 1: Install Docker and Docker Compose

Refer to the official [Docker installation docs](#) and [Compose plugin guide](#).

Step 2: Prepare Required Files

Place the license file and configuration JSONs in their respective directories as shown in the directory structure.

Step 3: Start the System

Navigate to your base folder and run the provided start.sh script:

```
1 ./start.sh
```

Docker will start all services and create the required networks and volumes.

To view running containers:

```
1 docker ps
```

To view logs for a container:

```
1 docker logs -f <container_name>
```

To shut down all services and clean up generated files, use the stop.sh script:

```
1 ./stop.sh
```

5. Docker Compose Setup

Below is the complete Docker Compose configuration for running PIES Studio in offline mode:

```
1 version: '3.7'
2
3 services:
4
5   # -----
6   # License Server
7   # -----
8   # Handles authentication, user, license, and SSO management.
9   pies-studio-license-server:
10    image: piesio/pies-studio-license-server:offline
11    container_name: pies-studio-license-server
12    ports:
13      - "9070:9070" # REST API exposed to internal services and web portals
14    environment:
15      - CONFIG_PATH=/bin/app/config/config.json # Path to the license service
16      - ENVIRONMENT=offline
17      - REDIRECT_URI_STUDIO=http://localhost:4200/login # OAuth redirect
18      - REDIRECT_URI_ADMIN=http://localhost:4100/auth/login
19      - ORG_DOMAIN=localhost
20    volumes:
21      # Read-only config file used at startup
22      - ./config/license/config.json:/bin/app/config/config.json:ro
23
24      # Encrypted license file mounted read-write for runtime updates
25      - ./config/license/pies_studio.license:/bin/app/pies_studio.license:rw
26    networks:
27      - pies-network
```

```

28     depends_on:
29         - redis
30         - mongo
31
32     # -----
33     # License Admin Portal
34     # -----
35     # A static admin UI (Angular) to manage user access and licenses.
36     pies-studio-license-web:
37         image: piesio/pies-studio-license-web:offline
38         container_name: pies-studio-license-web
39         ports:
40             - "4100:80"
41         volumes:
42             # Inject runtime config for the admin portal (config.web.json)
43             -
./config/license/config.web.json:/usr/share/nginx/html/browser/assets/env/config.json:
ro
44         networks:
45             - pies-network
46         depends_on:
47             - pies-studio-license-server
48
49     # -----
50     # Core Backend Server
51     # -----
52     # Hosts APIs for screens, workflows, apps, users and internal logic.
53     pies-studio-core:
54         image: piesio/pies-studio-core:offline
55         container_name: pies-studio-core
56         ports:
57             - "8080:8080"    # Public API port
58             - "9081:9081"    # Optional internal communication (e.g., pub/sub)
59         environment:
60             - CONFIG_PATH=/dist/config/config.json
61             - ENVIRONMENT=offline
62         volumes:
63             # Backend configuration for database, license service, etc.
64             - ./config/core/config.json:/dist/config/config.json:ro
65         networks:
66             - pies-network
67         depends_on:
68             - pies-studio-license-server
69             - mongo
70
71     # -----
72     # Studio Frontend Portal
73     # -----
74     # The no-code interface for end users to build apps.
75     pies-studio-web:
76         image: piesio/pies-studio-web:offline
77         container_name: pies-studio-web
78         ports:
79             - "4200:80"
80         volumes:
81             # Runtime environment config injected at container start
82             - ./config/web/config.json:/usr/share/nginx/html/assets/env/config.json:ro
83         networks:

```

```

84     - pies-network
85     depends_on:
86     - pies-studio-core
87
88     # -----
89     # Code Generation Engine
90     # -----
91     # Converts app models into actual deployable code and artifacts.
92     pies-studio-codegen:
93     image: piesio/pies-studio-codegen:offline
94     container_name: pies-studio-codegen
95     ports:
96     - "9090:9090"
97     environment:
98     - CONFIG_PATH=/loki/config/config.json
99     - ENVIRONMENT=offline
100    - MODE=listen
101    - DOCKER_HOST=tcp://docker:2376
102    - DOCKER_CERT_PATH=/certs/client/
103    - DOCKER_TLS_VERIFY=enable
104    volumes:
105    # Client certificates used to connect securely to Docker-in-Docker
106    - ./certs/client:/certs/client
107
108    # Shared folder for generated apps and temporary files
109    - ./generated:/loki/generated
110
111    # Codegen engine configuration file
112    - ./config/codegen/config.json:/loki/config/config.json:ro
113
114    # Vault unseal response for unlocking secrets on boot
115    - ./config/vault/config/init-response.json:/vault/config/init-response.json
116    networks:
117    - pies-network
118    depends_on:
119    - pies-studio-vault
120    - mysql
121    - redis
122    - docker
123
124    # -----
125    # Docker-in-Docker Service
126    # -----
127    # Allows PIES Studio to build and preview apps in isolation.
128    docker:
129    image: piesio/pies-studio-dind:offline
130    container_name: docker
131    environment:
132    - CONFIG_PATH=/loki/config/config.json
133    - ENVIRONMENT=offline
134    - DOCKER_TLS_CERTDIR=/certs/
135    - DOCKER_TLS_VERIFY=enable
136    - MODE=proxy
137    ports:
138    - "9010:9010" # HTTP port used by internal Docker proxy
139    - "9020:9020" # Optional gRPC or Docker events port
140    volumes:
141    - ./certs/client:/certs/client

```

```

142     - ./generated:/loki/generated
143     - ./config/codegen/config.json:/loki/config/config.json:ro
144     - ./config/vault/config/init-response.json:/vault/config/init-response.json
145 networks:
146     - pies-network
147 depends_on:
148     - pies-studio-vault
149 privileged: true # Required for running Docker inside Docker
150
151 # -----
152 # Preview Client
153 # -----
154 # UI for previewing apps deployed by the codegen engine.
155 pies-studio-preview:
156     image: piesio/pies-studio-preview:offline
157     container_name: pies-studio-preview
158     ports:
159         - "4300:80"
160     volumes:
161         # Preview environment configuration
162         - ./config/preview/config.json:/usr/share/nginx/html/assets/env/config.json:ro
163     networks:
164         - pies-network
165     depends_on:
166         - pies-studio-core
167         - pies-studio-codegen
168
169 # -----
170 # AI Engine (Optional)
171 # -----
172 # Handles AI-driven features (e.g., natural language generation).
173 pies-studio-ai:
174     image: piesio/pies-studio-ai:offline
175     container_name: pies-studio-ai
176     ports:
177         - "9075:9075"
178         - "9076:9076"
179     environment:
180         - AUTH_URL=http://host.docker.internal:9070/auth/key # Replace if using custom
host
181     networks:
182         - pies-network
183
184 # -----
185 # Vault (Secrets Storage)
186 # -----
187 # Used to store and retrieve sensitive secrets.
188 pies-studio-vault:
189     image: piesio/pies-studio-vault:offline
190     container_name: pies-studio-vault
191     ports:
192         - "8200:8200"
193     volumes:
194         # Vault bootstrap config (sealed initially)
195         - ./config/vault/config:/vault/config/
196
197         # Encrypted secrets and data
198         - ./config/vault/data:/vault/data:rw

```

```

199     networks:
200         - pies-network
201
202     # -----
203     # MySQL Database
204     # -----
205     # Required for preview containers
206     mysql:
207         image: mysql:8.0
208         container_name: mysql
209         ports:
210             - "3306:3306"
211         environment:
212             - MYSQL_ROOT_PASSWORD=<YOUR_MYSQL_DB_PASSWORD> # CHANGE THIS!
213             - MYSQL_DATABASE=pies_preview_db
214         networks:
215             - pies-network
216
217     # -----
218     # MongoDB (Primary Database)
219     # -----
220     # Used by all core and license services for app/user/storage.
221     mongo:
222         image: mongo
223         container_name: mongo
224         ports:
225             - "28018:27017" # Maps internal port 27017 to external 28018
226         environment:
227             - MONGO_INITDB_ROOT_USERNAME=root
228             - MONGO_INITDB_ROOT_PASSWORD=<YOUR_MONGO_DB_PASSWORD> # CHANGE THIS!
229         volumes:
230             - ./db/mongo:/data/db
231         networks:
232             - pies-network
233
234     # -----
235     # Redis (In-Memory Store)
236     # -----
237     # Used for caching sessions, rate limits, and pub/sub.
238     redis:
239         image: redis
240         container_name: redis
241         ports:
242             - "6379:6379"
243         networks:
244             - pies-network
245
246     # Shared network for all services
247     networks:
248         pies-network:
249             driver: bridge

```

PIES Studio Images

The following Docker images are used in this deployment. All are hosted under the piesio organization on DockerHub:

Service Name	Description	Port(s)	Image URL
pies-studio-web	No-code studio frontend	4200	
pies-studio-core	Backend API server	8080, 9081	
pies-studio-license-server	License and Auth management server	9070	
pies-studio-license-web	Admin portal for licenses and SSO	4100	
pies-studio-codegen	Code generation engine	9090	
docker	Docker-in-Docker service for preview execution	9010, 9020	
pies-studio-preview	Frontend client for previewing deployed apps	4300	
pies-studio-ai	Backend AI assistant service	9075, 9076	
pies-studio-vault	Modified Vault image for secure secret storage	8200	
mongo	MongoDB database	28018 (27017)	
mysql	MySQL database for codegen	3306	
redis	Caching and OTP queue	6379	

The setup also includes official images for:

- MySQL 8.0
- MongoDB
- Redis

Start and Stop Scripts

start.sh


```
1 docker compose up -d
```

stop.sh

```
1 docker compose down
2 rm -rf certs/client/*
3 rm -rf generated/*
```


6. Post-Setup Verification

After the services are up:

- Visit <http://localhost:4200> to access the PIES Studio portal.
- Visit <http://localhost:4100> to access the License Admin portal.
- Check logs of the codegen container to verify Vault unsealing.
- Use `docker exec -it mongo mongosh` to verify DB connection if needed.

If any containers fail, inspect logs using `docker logs` or use `docker-compose down` and retry after fixing config issues.

7. Running on Other Platforms (Kubernetes, Nomad, etc.)

 While this guide focuses on Docker Compose, the PIES Studio Offline stack is fully containerised and can be deployed on any container orchestration platform, including Kubernetes, Nomad, Docker Swarm or custom container runtimes.

To do so:

- Use the full `docker-compose.yml` as a reference for service definitions, ports, inter-container networking, and volume mounts.
 - Each container must have access to its corresponding `config.json` file through a volume mount or secret.
 - The `pies_studio.license` file must be mounted read-only into the license server container.
 - Vault must be unsealed via the shared volume between `pies-studio-vault` and `pies-studio-codegen` as specified in the compose file.
 - Expose ports using Ingress (Kubernetes) or equivalent Service / Proxy mechanisms.
 - Containers must run within the same network/namespace for internal communication.
 - You may use environment-specific mechanisms to inject secrets (e.g., Kubernetes Secrets, AWS SSM, Nomad Vault integrations).
-

8. Individual Installation

For details regarding installing each of the images individually, please refer to the 'Repository Overview' section on the DockerHub website for each of the images.

9. Support

For licensing issues or deployment support, please contact your onboarding specialist or reach out to support@pies.io.

Please include the following when raising an issue:

- Docker Compose logs (docker-compose logs)
 - Environment details (OS, Docker version, etc.)
-