

# Docker Hub 푸시 가이드

## <h2>목표</h2>

<p>Docker 이미지를 빌드하고 <code>stonesteel84/new-mcp-server</code> 레포지토리에 푸시하기</p>

## <h2>사전 준비사항</h2>

<ul>

- <li>Docker 설치 확인</li>
- <li>Docker Hub 계정 로그인</li>
- <li>환경 변수 설정</li>

</ul>

## <h2>단계별 실행 과정</h2>

<div class="step">

<h3>1단계: 환경 변수 확인</h3>

<pre><code># 현재 환경 변수 확인

```
echo "GEMINI_API_KEY: ${GEMINI_API_KEY:0:10}..." echo "NEXT_PUBLIC_SUPABASE_URL:
$NEXT_PUBLIC_SUPABASE_URL" echo "NEXT_PUBLIC_SUPABASE_ANON_KEY:
${NEXT_PUBLIC_SUPABASE_ANON_KEY:0:20}..."
```

필수 환경 변수:

- `GEMINI_API_KEY` : Gemini API 키
- `LLM_MODEL` : 사용할 모델 (기본값: gemini-2.0-flash)
- `NEXT_PUBLIC_SUPABASE_URL` : Supabase 프로젝트 URL
- `NEXT_PUBLIC_SUPABASE_ANON_KEY` : Supabase 익명 키

<div class="step">

<h3>2단계: Docker Hub 로그인</h3>

<pre><code># Docker Hub에 로그인

docker login

## 또는 사용자명 직접 지정

docker login -u stonesteel84

실행 결과 예시:

```
Username: stonesteel84
Password: *****
Login Succeeded
```

```
<div class="step">
  <h3>3단계: Docker 이미지 빌드</h3>
  <pre><code># 프로젝트 루트 디렉토리에서 실행
```

```
cd /path/to/ai-chat-hands-on-feature-mcp-client-3
```

## 이미지 빌드 (빌드 인자 포함)

```
docker build
--build-arg NEXT_PUBLIC_SUPABASE_URL="${NEXT_PUBLIC_SUPABASE_URL}"
--build-arg NEXT_PUBLIC_SUPABASE_ANON_KEY="${NEXT_PUBLIC_SUPABASE_ANON_KEY}"
-t stonesteel84/new-mcp-server:latest
-t stonesteel84/new-mcp-server:v1.0.0
.
```

### 빌드 과정 설명:

- `--build-arg`: 빌드 시 필요한 환경 변수 전달
- `-t`: 이미지 태그 지정 (latest와 버전 태그 모두 지정)
- `.`: 현재 디렉토리를 빌드 컨텍스트로 사용

예상 빌드 시간: 5-10분 (첫 빌드 시)

### 빌드 성공 확인:

```
docker images | grep stonesteel84/new-mcp-server
```

### 출력 예시:

| REPOSITORY                  | TAG    | IMAGE ID     | CREATED       | SIZE  |
|-----------------------------|--------|--------------|---------------|-------|
| stonesteel84/new-mcp-server | latest | abc123def456 | 2 minutes ago | 450MB |
| stonesteel84/new-mcp-server | v1.0.0 | abc123def456 | 2 minutes ago | 450MB |

```
<div class="step">
  <h3>4단계: 이미지 태그 확인</h3>
  <pre><code># 빌드된 이미지 목록 확인
```

docker images stonesteel84/new-mcp-server

```
<div class="step">
  <h3>5단계: Docker Hub에 푸시</h3>
  <pre><code># latest 태그 푸시
```

docker push stonesteel84/new-mcp-server:latest

## 버전 태그 푸시

docker push stonesteel84/new-mcp-server:v1.0.0

### 푸시 과정 설명:

1. 이미지 레이어 압축 및 업로드
2. Docker Hub에 메타데이터 전송
3. 레포지토리에 이미지 등록

예상 푸시 시간: 3-10분 (이미지 크기에 따라)

### 푸시 진행 상황 예시:

```
The push refers to repository [docker.io/stonesteel84/new-mcp-server]
abc123def456: Pushing [=====] 450MB
def456ghi789: Pushing [=====] 120MB
...
latest: digest: sha256:abc123... size: 1234
```

```
<div class="step">
  <h3>6단계: 푸시 확인</h3>
  <h4>방법 1: Docker Hub 웹사이트 확인</h4>
  <ol>
    <li>https://hub.docker.com 접속</li>
    <li>로그인 후 <code>stonesteel84/new-mcp-server</code> 레포지토리 확인</li>
    <li>이미지가 업로드되었는지 확인</li>
  </ol>
  <h4>방법 2: 명령어로 확인</h4>
  <pre><code># 이미지 정보 확인
```

```
docker inspect stonesteel84/new-mcp-server:latest
```

## 원격 레포지토리에서 pull 테스트

```
docker pull stonesteel84/new-mcp-server:latest
```

<h2>전체 명령어 한번에 실행</h2>

<h3>Linux/Mac (Bash)</h3>

```
<pre><code>#!/bin/bash
```

## 환경 변수 확인

```
echo "=== 환경 변수 확인 ===" if [ -z "$GEMINI_API_KEY" ] || [ -z "$NEXT_PUBLIC_SUPABASE_URL" ] || [ -z "$NEXT_PUBLIC_SUPABASE_ANON_KEY" ]; then echo "❌ 필수 환경 변수가 설정되지 않았습니다!" exit 1 fi
echo "✅ 환경 변수 확인 완료"
```

## Docker Hub 로그인

```
echo "=== Docker Hub 로그인 ===" docker login -u stonesteel84 if [ $? -ne 0 ]; then echo "❌ Docker Hub 로그인 실패" exit 1 fi echo "✅ 로그인 성공"
```

## 이미지 빌드

```
echo "=== Docker 이미지 빌드 ===" docker build
--build-arg NEXT_PUBLIC_SUPABASE_URL="${NEXT_PUBLIC_SUPABASE_URL}"
--build-arg NEXT_PUBLIC_SUPABASE_ANON_KEY="${NEXT_PUBLIC_SUPABASE_ANON_KEY}"
-t stonesteel84/new-mcp-server:latest
-t stonesteel84/new-mcp-server:v1.0.0
.
```

```
if [ $? -ne 0 ]; then echo "❌ 이미지 빌드 실패" exit 1 fi echo "✅ 이미지 빌드 완료"
```

## 이미지 푸시

```
echo "=== Docker Hub에 푸시 ===" docker push stonesteel84/new-mcp-server:latest docker push
stonesteel84/new-mcp-server:v1.0.0
```

```
if [ $? -ne 0 ]; then echo "❌ 이미지 푸시 실패" exit 1 fi echo "✅ 푸시 완료!"
```

```
echo "" echo "🎉 모든 작업이 완료되었습니다!" echo "📦 이미지: stonesteel84/new-mcp-server:latest" echo
"🌐 확인: https://hub.docker.com/r/stonesteel84/new-mcp-server"
```

```
<h3>Windows (PowerShell)</h3>
<pre><code># 환경 변수 확인

```

Write-Host "=== 환경 변수 확인 ===" -ForegroundColor Cyan if (-not \$env:GEMINI\_API\_KEY -or -not \$env:NEXT\_PUBLIC\_SUPABASE\_URL -or -not \$env:NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY) { Write-Host "❌ 필수 환경 변수가 설정되지 않았습니다!" -ForegroundColor Red exit 1 } Write-Host "✅ 환경 변수 확인 완료" -ForegroundColor Green

## Docker Hub 로그인

Write-Host "=== Docker Hub 로그인 ===" -ForegroundColor Cyan docker login -u stonesteel84 if (\$LASTEXITCODE -ne 0) { Write-Host "❌ Docker Hub 로그인 실패" -ForegroundColor Red exit 1 } Write-Host "✅ 로그인 성공" -ForegroundColor Green

## 이미지 빌드

Write-Host "=== Docker 이미지 빌드 ===" -ForegroundColor Cyan docker build --build-arg NEXT\_PUBLIC\_SUPABASE\_URL="\$env:NEXT\_PUBLIC\_SUPABASE\_URL" --build-arg NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY="\$env:NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY" -t stonesteel84/new-mcp-server:latest -t stonesteel84/new-mcp-server:v1.0.0 `.

if (\$LASTEXITCODE -ne 0) { Write-Host "❌ 이미지 빌드 실패" -ForegroundColor Red exit 1 } Write-Host "✅ 이미지 빌드 완료" -ForegroundColor Green

## 이미지 푸시

Write-Host "=== Docker Hub에 푸시 ===" -ForegroundColor Cyan docker push stonesteel84/new-mcp-server:latest docker push stonesteel84/new-mcp-server:v1.0.0

if (\$LASTEXITCODE -ne 0) { Write-Host "❌ 이미지 푸시 실패" -ForegroundColor Red exit 1 } Write-Host "✅ 푸시 완료!" -ForegroundColor Green

Write-Host "" Write-Host "🎉 모든 작업이 완료되었습니다!" -ForegroundColor Green Write-Host "📦 이미지: stonesteel84/new-mcp-server:latest" Write-Host "🌐 확인: <https://hub.docker.com/r/stonesteel84/new-mcp-server>"

```
<h2>문제 해결</h2>
```

```
<div class="warning">
  <h3>1. 빌드 실패: 환경 변수 누락</h3>
  <pre><code># 해결: 환경 변수 설정 확인

```

```
echo $GEMINI_API_KEY echo $NEXT_PUBLIC_SUPABASE_URL
```

```
<div class="warning">
  <h3>2. 로그인 실패</h3>
  <pre><code># 해결: 토큰 사용 (권장)
```

```
docker login -u stonesteel84 --password-stdin < ~/docker-token.txt
```

```
<div class="warning">
  <h3>3. 푸시 실패: 권한 없음</h3>
  <pre><code># 해결: Docker Hub에서 레포지토리 생성 확인
```

## <https://hub.docker.com/repositories> 에서 확인

---

```
</div>

<div class="warning">
  <h3>4. 네트워크 오류</h3>
  <pre><code># 해결: 프록시 설정 또는 재시도
```

```
docker push stonesteel84/new-mcp-server:latest --retry 3
```

```
<h2>이미지 사용 방법</h2>
<p><strong>다른 서버에서 이미지 사용:</strong></p>
<pre><code># 이미지 pull
```

```
docker pull stonesteel84/new-mcp-server:latest
```

## 컨테이너 실행

---

```
docker run -d
-p 4000:4000
-e GEMINI_API_KEY=your_key
-e LLM_MODEL=gemini-2.0-flash
-e NEXT_PUBLIC_SUPABASE_URL=your_url
-e NEXT_PUBLIC_SUPABASE_ANON_KEY=your_key
--name new-mcp-server
stonesteel84/new-mcp-server:latest
```

```
<h2>체크리스트</h2>
<ul class="checklist">
  <li>Docker 설치 확인</li>
  <li>Docker Hub 계정 로그인</li>
  <li>환경 변수 설정 확인</li>
  <li>이미지 빌드 성공</li>
  <li>이미지 푸시 성공</li>
  <li>Docker Hub에서 이미지 확인</li>
</ul>

<h2>참고 링크</h2>
<ul>
  <li>Docker Hub: <a href="https://hub.docker.com/r/stonesteel84/new-mcp-server">https://hub.docker.com/r/stonesteel84/new-mcp-server</a></li>
  <li>Docker 문서: <a href="https://docs.docker.com/">https://docs.docker.com/</a></li>
  <li>Next.js Docker 가이드: <a href="https://nextjs.org/docs/deployment#docker-image">https://nextjs.org/docs/deployment#docker-image</a></li>
</ul>

<hr>
<p style="text-align: center; color: #666; margin-top: 40px;">
  <strong>작성일:</strong> 2025-11-28<br>
  <strong>버전:</strong> 1.0.0
</p>

<script>
  // PDF로 저장 안내
  window.onload = function() {
    console.log('PDF로 저장하려면: Ctrl+P (또는 Cmd+P) → "PDF로 저장" 선택');
  }
</script>
```