

Cài đặt Chat Channels cho OpenClaw

Sau khi cài đặt chúng ta đã có thể chat với OpenClaw qua giao diện web. Nhưng để thuận tiện OpenClaw cung cấp rất nhiều các kênh chat khác nhau như Telegram, Discord, Zalo...

- [Cài Discord Bot cho OpenClaw](#)
- [Cài Telegram Bot cho OpenClaw](#)

Cài Discord Bot cho OpenClaw

Mục tiêu

Cấu hình Discord Bot để chat với OpenClaw qua DM.

Kết quả sau khi cài:

- Có một Bot Discord online.
- Có thể DM và nhận phản hồi từ Bot.

Điều kiện trước khi làm

OpenClaw Gateway đã chạy bằng Docker/Portainer.

Đã cài đặt backend bất kỳ, có thể chat với OpenClaw thông qua web hoặc nền tảng khác.

Tóm tắt nhanh

Các bước chính:

1. Tạo Discord Application
2. Tạo Bot và copy Bot Token
3. Bật Message Content Intent + Server Members Intent
4. Cấu hình Installation → Guild Install → bot + applications.commands
5. Invite bot vào server riêng
6. Thêm DISCORD_BOT_TOKEN vào Portainer
7. Cài plugin @openclaw/discord
8. Bật channels.discord trong OpenClaw config
9. Restart OpenClaw Gateway
10. Pair Discord DM với OpenClaw
11. Chat với OpenClaw qua DM bot

Tạo Discord Application và Bot

Vào [Discord Developer Portal](#) > New Application > Đặt tên (Ex: OClaw)

Ở menu bên trái chọn **Bot** > Reset Token > Copy Bot Token (*lưu lại token này để dùng sau*)

Bật Gateway Intents cho Bot

Vẫn trong tab **Bot** > Privileged Gateway Intents

- Message Content Intent: ON → Cho phép bot đọc nội dung tin nhắn.
- Server Members Intent: ON → Hữu ích nếu sau này muốn allowlist theo user/role/server.
- Presence Intent: OFF

Save Changes

Cấu hình Installation

Chuyển sang tab **Installation**

Installation Contexts

- Guild Install: ON
- User Install: optional

Install Link

- Chọn Discord Provided Link

Default Install Settings

- Với Guild Install, thêm scopes:
 - bot
 - applications.commands

Khi chọn bot, Discord sẽ hiện phần quyền cho bot. Chọn tối thiểu:

- View Channels
- Send Messages
- Read Message History
- Embed Links
- Attach Files
- Add Reactions

Nếu muốn dùng thread/forum channel, bật thêm:

- `Send Messages in Threads`

Sau đó copy **Install Link**, mở bằng trình duyệt và mời bot vào server riêng của bạn.

Bật Developer Mode và lấy ID

Trong Discord app > User Settings > Advanced > Developer Mode: ON

Lấy các ID cần thiết:

- Right-click server icon → Copy Server ID
- Right-click avatar của bạn → Copy User ID

Các giá trị cần có:

- `DISCORD_BOT_TOKEN=...` // → Không gửi cho người khác.
- `DISCORD_SERVER_ID=...` // → Dùng nếu sau này muốn cấu hình bot trong server/channel.
- `DISCORD_USER_ID=...` // → Dùng để kiểm soát ai được phép chat với bot.

Thêm Discord Bot Token vào Portainer stack

Vào Portainer > Stacks > openclaw > Editor

Trong phần Environment variables, thêm:

- `DISCORD_BOT_TOKEN=token_bot_discord_c?a_b?n`

Trong compose, ở service openclaw-gateway, thêm vào phần environment:

- `DISCORD_BOT_TOKEN: ${DISCORD_BOT_TOKEN}`

Bấm **Update the stack**

Test Discord Bot Token

Chạy:

```
docker exec -it openclaw-gateway sh -lc '
node -e "
fetch("https://discord.com/api/v10/users/@me", {
  headers: { Authorization: \"Bot \" + process.env.DISCORD_BOT_TOKEN }
})
.then(async r => {
  console.log(\"status\", r.status);
  console.log(await r.text());
})
.catch(e => {
  console.error(e);
  process.exit(1);
});
"
```

Kết quả mong muốn:

```
status 200
{"id":"...","username":"...","bot":true,...}
```

Nếu ra:

```
status 401
```

thì token sai hoặc đã bị reset.

Cài plugin Discord cho OpenClaw

Chỉ cấu hình `channels.discord` là chưa đủ. OpenClaw cần plugin Discord.

Kiểm tra trạng thái channel:

```
docker exec -it openclaw-gateway \
node dist/index.js channels list --all | grep -i discord
```

Nếu thấy:

```
Discord: not installed, configured, disabled, run openclaw plugins install @openclaw/discord
```

thì cài plugin:

```
docker exec -it openclaw-gateway \
node dist/index.js plugins install @openclaw/discord
```

Sau đó kiểm tra lại:

```
docker exec -it openclaw-gateway \
node dist/index.js channels list --all | grep -i discord
```

Bật Discord channel trong OpenClaw config

Ghi config Discord:

```
docker exec -it openclaw-gateway \
node dist/index.js config set --batch-json '[
  {
    "path": "channels.discord.enabled",
    "value": true
  },
  {
    "path": "channels.discord.token",
    "value": {
      "source": "env",
      "provider": "default",
      "id": "DISCORD_BOT_TOKEN"
    }
  },
  {
    "path": "channels.discord.groupPolicy",
    "value": "allowlist"
  }
]
```

Kiểm tra config:

```
docker exec -it openclaw-gateway \
node dist/index.js config get channels.discord --json
```

Kết quả mong muốn:

```
{
  "enabled": true,
  "token": {
    "source": "env",
    "provider": "default",
    "id": "DISCORD_BOT_TOKEN"
  },
  "groupPolicy": "allowlist"
}
```

Restart gateway:

```
docker restart openclaw-gateway
```

Bot trên Discord nên chuyển sang **Online** là thành công.

Pair Discord DM với OpenClaw

Mở DM với bot Discord, gửi `hello` hoặc `pair` > Bot sẽ trả về pairing code.

Trên Docker host, kiểm tra pairing request:

```
docker exec -it openclaw-gateway \
node dist/index.js pairing list discord
```

Approve pairing code:

```
docker exec -it openclaw-gateway \
node dist/index.js pairing approve discord <CODE>
```

Sau đó quay lại Discord DM và test:

```
B?n ?ang ho?t ??ng ch?a? Tr? l?i ng?n g?n.
```

Nếu bot trả lời được, Discord DM đã hoạt động.

Phụ lục

Bot v?n Offline

Kiểm tra:

```
docker exec -it openclaw-gateway \
node dist/index.js channels list --all | grep -i discord
```

Nếu thấy `not installed`, cài plugin:

```
docker exec -it openclaw-gateway \
node dist/index.js plugins install @openclaw/discord
```

Restart:

```
docker restart openclaw-gateway
```

Token r?ng trong container

Kiểm tra:

```
docker exec -it openclaw-gateway sh -lc '
if [ -n "$DISCORD_BOT_TOKEN" ]; then
  echo "OK"
else
  echo "EMPTY"
fi
'
```

Nếu ra `EMPTY`, kiểm tra lại Portainer Environment variables và compose `environment:`.

Token sai

Test:

```
docker exec -it openclaw-gateway sh -lc '
node -e "
fetch("https://discord.com/api/v10/users/@me", {
  headers: { Authorization: \"Bot \" + process.env.DISCORD_BOT_TOKEN }
})
.then(async r => {
  console.log(\"status\", r.status);
  console.log(await r.text());
})
"
'
```

Nếu ra 401, reset token trong Discord Developer Portal rồi cập nhật lại Portainer.

Bot online nh?ng không ph?n h?i DM

Kiểm tra pairing:

```
docker exec -it openclaw-gateway \
node dist/index.js pairing list discord
```

Approve pairing code nếu có:

```
docker exec -it openclaw-gateway \
node dist/index.js pairing approve discord <CODE>
```

Xem log Discord:

```
docker exec -it openclaw-gateway \
node dist/index.js channels logs --channel discord --lines 200
```

Cài Telegram Bot cho OpenClaw

Mục tiêu

Cấu hình Telegram Bot để chat với OpenClaw qua DM.

Luồng hoạt động:

- Bạn
- → Telegram DM Bot
- → OpenClaw Gateway
- → Model
- → trả lời lại Telegram

Tóm tắt nhanh

Các bước chính:

1. Tạo bot bằng @BotFather
2. Copy Telegram Bot Token
3. Thêm TELEGRAM_BOT_TOKEN vào Portainer
4. Cài plugin @openclaw/telegram nếu chưa có
5. Bật channels.telegram
6. Restart OpenClaw Gateway
7. DM bot để lấy pairing code
8. Approve pairing code
9. Chat với OpenClaw qua Telegram DM

Tạo Telegram Bot

Mở Telegram, tìm: > gửi

Làm theo hướng dẫn:

- Đặt tên hiển thị cho bot
- Đặt username cho bot, bắt buộc kết thúc bằng bot

Sau khi tạo xong, BotFather sẽ trả về token dạng `1234567890:ABCDEF_XXXXXXXXXXXXXXXXXXXX`

Lưu token lại. Không gửi token cho người khác.

Thêm token vào Portainer Stack

Portainer → Stacks → openclaw → Editor

Trong phần Environment variables, thêm:

- `TELEGRAM_BOT_TOKEN=token_telegram_bot_c?a_b?n`

Trong compose, thêm vào service openclaw-gateway.environment:

- `TELEGRAM_BOT_TOKEN: ${TELEGRAM_BOT_TOKEN}`

Update the stack.

Kiểm tra token trong container

Chạy:

```
docker exec -it openclaw-gateway sh -lc '
if [ -n "$TELEGRAM_BOT_TOKEN" ]; then
  echo "OK: TELEGRAM_BOT_TOKEN is set"
else
  echo "ERROR: TELEGRAM_BOT_TOKEN is empty"
fi
'
```

Test token với Telegram API:

```
docker exec -it openclaw-gateway sh -lc '
node -e "
fetch("https://api.telegram.org/bot\" + process.env.TELEGRAM_BOT_TOKEN + \"/getMe\")
.then(async r => {
  console.log(\"status\", r.status);
  console.log(await r.text());
})
.catch(e => {
  console.error(e);
  process.exit(1);
});
"
```

Kết quả đúng:

```
status 200
{"ok":true,...}
```

Nếu `ok:false` hoặc lỗi `401`, token sai hoặc đã bị reset.

Cài plugin Telegram

Kiểm tra trạng thái Telegram channel:

```
docker exec -it openclaw-gateway \
node dist/index.js channels list --all | grep -i telegram
```

Nếu thấy `not installed`, cài plugin:

```
docker exec -it openclaw-gateway \
node dist/index.js plugins install @openclaw/telegram
```

Kiểm tra lại:

```
docker exec -it openclaw-gateway \
node dist/index.js channels list --all | grep -i telegram
```

Bật Telegram channel

Ghi config Telegram:

```
docker exec -it openclaw-gateway \
node dist/index.js config set --batch-json '[
  {
    "path": "channels.telegram.enabled",
    "value": true
  },
  {
    "path": "channels.telegram.dmPolicy",
    "value": "pairing"
  },
  {
    "path": "channels.telegram.groupPolicy",
    "value": "allowlist"
  }
]'
```

Ý nghĩa:

```
// Ng??i nh?n DM l?n ??u c?n pairing code.
dmPolicy = pairing

// Không cho bot ho?t ??ng t? do trong group n?u ch?a c?u hình.
groupPolicy = allowlist
```

Restart OpenClaw

```
docker restart openclaw-gateway
```

Pair Telegram DM

Mở Telegram, nhấn cho bot `/start` hoặc `hello`

Bot sẽ trả về pairing code.

Trên Docker host, xem pairing request:

```
docker exec -it openclaw-gateway \  
node dist/index.js pairing list telegram
```

Approve code:

```
docker exec -it openclaw-gateway \  
node dist/index.js pairing approve telegram <CODE>
```

Ví dụ:

```
docker exec -it openclaw-gateway \  
node dist/index.js pairing approve telegram ABCD1234
```

Pairing code thường hết hạn sau khoảng 1 giờ.

Test chat

Trong Telegram DM với bot, gửi tin nhắn bất kỳ.

Nếu bot trả lời được, Telegram đã hoạt động.

Troubleshooting

Bot không tr? l?i

Kiểm tra channel:

```
docker exec -it openclaw-gateway \  
node dist/index.js channels list --all | grep -i telegram
```

Xem log channel:

```
docker exec -it openclaw-gateway \  
node dist/index.js channels logs --channel telegram --lines 200
```

Token r?ng

```
docker exec -it openclaw-gateway sh -lc '
if [ -n "$TELEGRAM_BOT_TOKEN" ]; then echo OK; else echo EMPTY; fi
'
```

Nếu ra `EMPTY`, kiểm tra lại Portainer Environment variables và compose.

Token sai

Test lại Telegram API:

```
docker exec -it openclaw-gateway sh -lc '
node -e "
fetch("https://api.telegram.org/bot\" + process.env.TELEGRAM_BOT_TOKEN + \"/getMe\")
.then(async r => {
  console.log(\"status\", r.status);
  console.log(await r.text());
})
"
'
```

Nếu response có lỗi `401`, tạo/reset token trong BotFather rồi cập nhật lại Portainer.

Bot tr? pairing code nh?ng approve xong v?n không tr? l?i

Kiểm tra pairing:

```
docker exec -it openclaw-gateway \
node dist/index.js pairing list telegram
```

Xem log:

```
docker logs --tail 160 openclaw-gateway | grep -iE "telegram|pairing|error"
```

