ChatGPT

BadAi: Pipeline Locale con ComfyUI in Docker (Guida Passo-Passo)

Overview: In questa guida configuriamo da zero *BadAi*, una pipeline locale per generare set di foto "da feed" usando ComfyUI e modelli open-source. Utilizzeremo Docker Desktop su Windows per creare un container Linux che esegue ComfyUI con tutti i componenti necessari: come modello base useremo **FLUX.1 [dev]** (open-weights) oppure **Stable Diffusion XL (SDXL)**; per mantenere **un'identità coerente** nelle immagini impiegheremo moduli come **PhotoMaker V2** o **InstantID**; per applicare uno **stile** specifico useremo **IP-Adapter**; per controllare **posa e coerenza fisica** integreremo **ControlNet OpenPose**; infine, per creare **brevi animazioni** da immagini statiche useremo **LivePortrait**. Ogni fase includerà istruzioni dettagliate e test per verificare il corretto funzionamento del sistema prima di passare allo step successivo.

Step 1: Installazione di Docker Desktop e Prerequisiti

1. **Installa Docker Desktop:** Scarica e installa **Docker Desktop per Windows** dal sito ufficiale Docker. Durante l'installazione, **abilita WSL2** come backend: Docker su Windows utilizza infatti il sottosistema Linux (WSL2) per eseguire i container. Assicurati di **abilitare WSL2 e il supporto GPU** nelle impostazioni se prevedi di usare la GPU NVIDIA del tuo sistema [1] . *(Nota: verifica che la Virtualizzazione sia attiva nel BIOS, poiché WSL2 ne ha bisogno.)*

2. **Abilita WSL2 (se non già fatto):** Se WSL2 non è ancora attivo sul tuo sistema, abilitalo seguendo le istruzioni Microsoft. In breve, apri PowerShell come amministratore ed esegui:

   ```
   wsl --install
   ```

   Questo comando abiliterà il **Windows Subsystem for Linux 2**. Al termine, riavvia il PC. (Non è necessario installare manualmente una distro Linux separata: Docker creerà automaticamente un ambiente Linux tramite WSL2.) [1]

3. **Driver NVIDIA aggiornati:** Se disponi di una **GPU NVIDIA** e intendi usarla nei container, installa o aggiorna i driver grafici all'ultima versione con supporto per WSL2. I driver recenti di NVIDIA per Windows includono il supporto CUDA in WSL2, necessario per consentire al container Docker di accedere alla GPU [2] .

4. **Installa Git:** Avremo bisogno di Git per clonare repository contenenti il codice e i modelli. Se non lo hai già, scarica e installa **Git per Windows** dal sito ufficiale (Git SCM). Questo ci permetterà di scaricare i file di progetto in seguito [3] .

5. **Verifica Docker in esecuzione:** Dopo l'installazione, verifica che Docker funzioni correttamente. Apri un **prompt dei comandi** (o PowerShell) e lancia il comando:

   ```
   docker run hello-world
   ```

Questo avvierà un container di test; dovresti vedere un messaggio di successo che conferma il corretto funzionamento di Docker (il messaggio “Hello from Docker!”). Se ottieni quell’output senza errori, Docker è installato ed eseguito correttamente sul tuo sistema.

6. **(Opzionale) Verifica accesso GPU nei container:** Se prevedi di usare la GPU per accelerare il pipeline (consigliato per modelli pesanti come SDXL o FLUX), fai un test rapido per assicurarti che Docker abbia accesso alla GPU. Esegui nel prompt:

   ```
   docker run --rm --gpus all nvidia/cuda:11.4.2-base nvidia-smi
   ```

   Questo comando avvia un container Linux base con CUDA e lancia l’utility `nvidia-smi` (che elenca le GPU NVIDIA disponibili). Dovresti vedere a schermo le informazioni della tua scheda video (nome modello, memoria, driver, ecc.) [4]. Se `nvidia-smi` mostra correttamente la GPU, significa che Docker **vede la GPU** e potremo usarla nei passi successivi. In caso contrario, ricontrolla di aver installato i driver giusti (vedi step precedente) e di aver abilitato l’integrazione della GPU in Docker Desktop.

   Test di fine Step 1: A questo punto dovresti avere Docker Desktop avviato su Windows, con WSL2 attivo. Il comando di test `hello-world` deve funzionare senza errori. Se hai una GPU NVIDIA e hai eseguito il test opzionale con `nvidia-smi` nel container, dovresti aver confermato che la GPU è accessibile. Se tutto è andato bene, sei pronto per procedere al passo successivo (installazione dell’ambiente ComfyUI nel container).

(Una volta completato lo Step 1 con successo, scrivi un breve feedback o eventuali errori riscontrati, e procediamo con lo Step 2.) [1] [2] [3] [4]

---

[1] GitHub - Kaouthia/ComfyUI-Docker: Dockerfile and Docker Compose setup for ComfyUI - Works on Linux or Windows for NVIDIA GPUs
https://github.com/Kaouthia/ComfyUI-Docker

[2] [3] Running ComfyUI in Docker on Windows or Linux - John Aldred
https://www.johnaldred.com/running-comfyui-in-docker-on-windows-or-linux/

[4] WSL 2 GPU Support for Docker Desktop on NVIDIA GPUs
https://www.docker.com/blog/wsl-2-gpu-support-for-docker-desktop-on-nvidia-gpus/