



# ICT Training Center

Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda



# **SPRING AI**

## **GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA**

---

**Simone Scannapieco**

Corso avanzato per Venis S.p.A, Venezia, Italia

**Novembre 2025**

# DOCKER MODEL RUNNER

---

- ➔ Risposta di Docker ad Ollama
  - ➔ LLM in Docker *container* locali
  - ➔ Modelli AI generici dockerizzabili (WIP)
  - ➔ Docker mette a disposizione una serie di modelli *open source* scaricabili tramite Engine o Desktop
- ➔ Requisiti: <https://www.ajeetraina.com/docker-model-runner-tutorial-and-cheatsheet-mac-windows-and-linux-support/>
- ➔ <https://docs.docker.com/ai/model-runner/>
- ➔ <https://www.docker.com/blog/run-llms-locally/>
- ➔ <https://www.docker.com/blog/introducing-docker-model-runner/>

- ➔ Tramite Docker Engine
- ➔ Tramite Docker Desktop

### Caricamento LLM in locale

```
docker model pull ai/gemma3
```

### Esecuzione LLM in locale

```
docker model run ai/gemma3
```

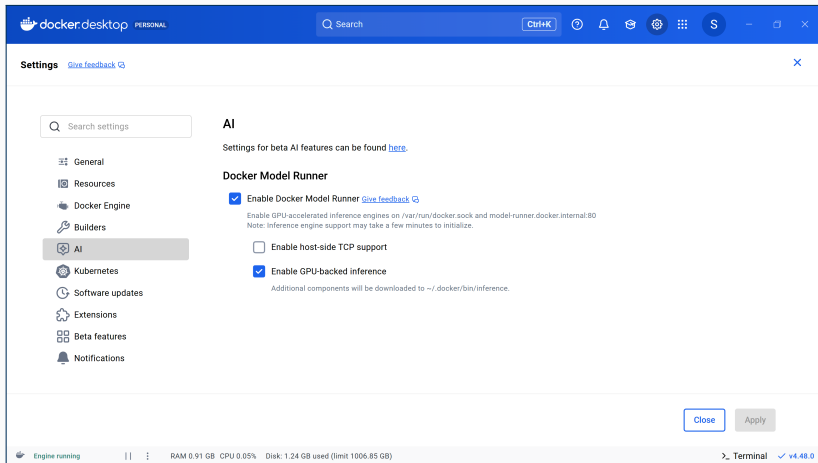
### Elenco LLM in locale

```
docker model list
```

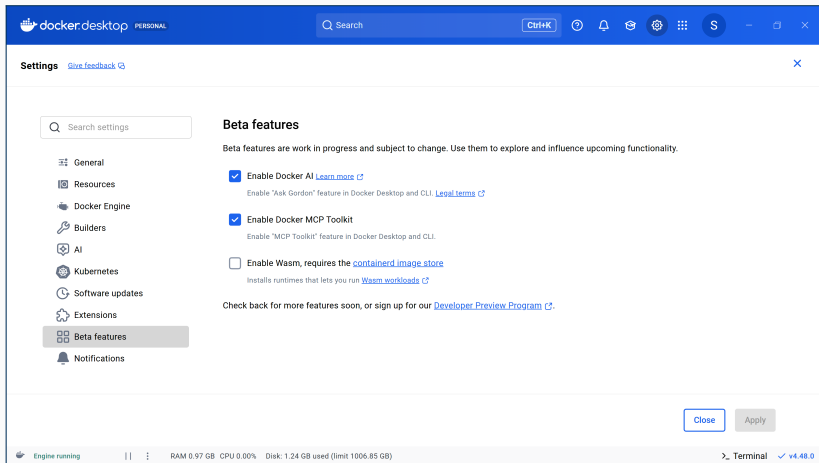
### Eliminazione LLM in locale

```
docker model rm ai/gemma3
```

### 1 Verificare le impostazioni relative ad AI



## 2 Verificare le impostazioni relative alle Beta features

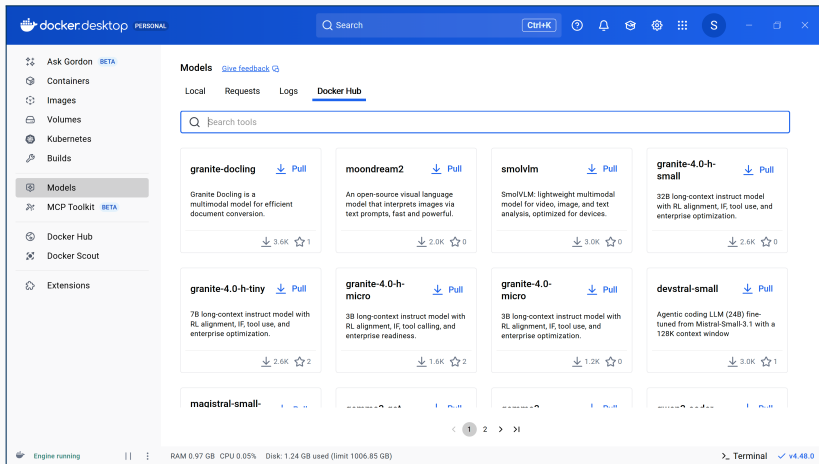




# DOCKER MODEL RUNNER

## UTILIZZO DI BASE - DOCKER DESKTOP

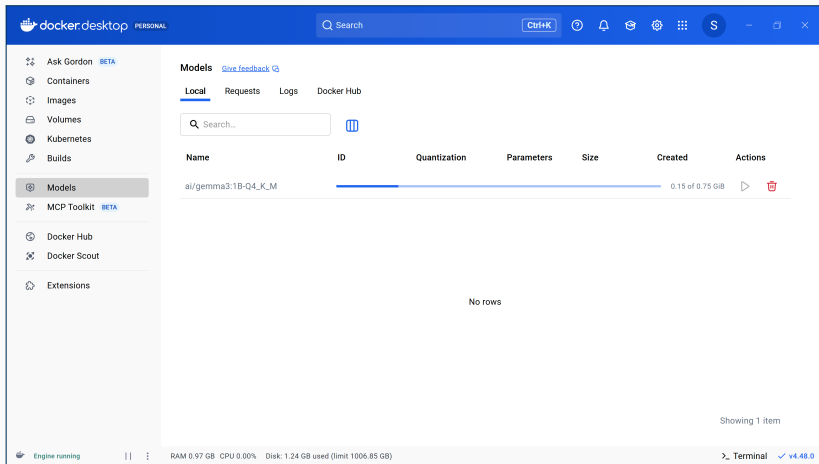
### 3 Accedere al pannello Docker Hub della sezione Models





The screenshot shows the Docker Desktop application window. The top bar is blue with the Docker logo, 'docker.desktop PERSONAL', a search bar, and a 'Ctrl+K' button. The left sidebar contains a list of navigation items: Ask Gordon (BETA), Containers, Images, Volumes, Kubernetes, Builds, Models (selected), MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main content area is titled 'Models' with a 'Give feedback' link. Below the title are tabs for 'Local', 'Requests', 'Logs', and 'Docker Hub' (selected). A search bar labeled 'Search tools' is present. The main area displays a grid of model cards, each with a name, a 'Pull' button, a description, and download statistics. The models shown are: granite-docling, moondream2, smolvlm, granite-4.0-h-small, granite-4.0-h-tiny, granite-4.0-h-micro, granite-4.0-micro, devstral-small, and maoqistral-small. The bottom status bar shows 'Engine running', system resources (RAM 0.97 GB, CPU 0.05%, Disk 1.24 GB used), and a 'Terminal' button with a version indicator 'v4.48.0'.

Model Name	Description	Download Size	Stars
granite-docling	Granite Docling is a multimodal model for efficient document conversion.	3.6K	1
moondream2	An open-source visual language model that interprets images via text prompts, fast and powerful.	2.0K	0
smolvlm	SmolVLM: lightweight multimodal model for video, image, and text analysis, optimized for devices.	3.0K	0
granite-4.0-h-small	32B long-context instruct model with RL alignment, IF, tool use, and enterprise optimization.	2.6K	0
granite-4.0-h-tiny	7B long-context instruct model with RL alignment, IF, tool use, and enterprise optimization.	2.6K	2
granite-4.0-h-micro	3B long-context instruct model with RL alignment, IF, tool calling, and enterprise readiness.	1.6K	2
granite-4.0-micro	3B long-context instruct model with RL alignment, IF, tool use, and enterprise optimization.	1.2K	0
devstral-small	Agentic coding LLM (24B) fine-tuned from Mistral-Small-3.1 with a 128K context window	3.0K	1
maoqistral-small			

### 4 Selezionare il modello ed eventuale versionamento quantizzato



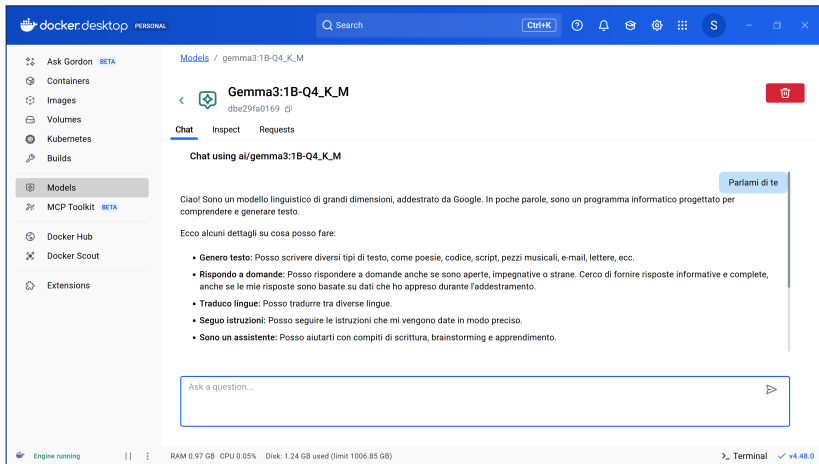
The screenshot shows the Docker Desktop interface with the 'Models' section selected in the left sidebar. The 'Local' tab is active, displaying a table of local models. The table has columns: Name, ID, Quantization, Parameters, Size, Created, and Actions. One model is listed: 'ai/gemma3:1B-Q4\_K\_M'. The 'Quantization' column for this model shows a progress bar indicating 0.15 of 0.75 GiB. The bottom status bar shows 'Engine running', 'RAM 0.97 GB', 'CPU 0.00%', and 'Disk: 1.24 GB used (limit 1006.85 GB)'. The bottom right corner shows 'Terminal' and 'v4.48.0'.

Name	ID	Quantization	Parameters	Size	Created	Actions
ai/gemma3:1B-Q4_K_M		<div></div>		0.15 of 0.75 GiB		 

No rows

Showing 1 item

### 5 Utilizzare il modello da linea di comando integrata



The screenshot shows the Docker Desktop application window. On the left is a sidebar with navigation options: Ask Gordon (BETA), Containers, Images, Volumes, Kubernetes, Builds, Models (selected), MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main panel displays the 'Models' section, specifically the 'Gemma3:1B-Q4\_K\_M' model (dbe29fa0169). Below the model name are tabs for 'Chat', 'Inspect', and 'Requests'. The 'Chat' tab is active, showing a chat interface with the title 'Chat using ai/gemma3:1B-Q4\_K\_M'. The chat content includes a greeting from the model and a list of capabilities: writing text, answering questions, translating, following instructions, and acting as an assistant. At the bottom of the chat area is a text input field with the placeholder 'Ask a question...' and a send button. The top of the window has a search bar and system icons. The bottom status bar shows 'Engine running', system resources (RAM 0.97 GB, CPU 0.05%, Disk 1.24 GB used), and a 'Terminal' button.

➔ Come fosse un servizio **OpenAI!**

### File pom.xml

```
...  
    <dependencies>  
        <dependency>  
            <groupId>org.springframework.boot</groupId>  
            <artifactId>spring-boot-starter-web</artifactId>  
        </dependency>  
        <dependency>  
            <groupId>org.springframework.ai</groupId>  
            <artifactId>spring-ai-starter-model-openai</artifactId>  
        </dependency>  
    </dependencies>  
...
```

### File application.yml

```
spring:  
  application:  
    name: demo  
  ai:  
    openai:  
      api-key: pippoplutopaperino  
      base-url: http://localhost:12434/engines  
      chat:  
        options:  
          model: ai/gemma3
```