

S3LLM Docker Tutorial

To follow this tutorial, ensure you have the latest versions of Docker and Nvidia CUDA installed.

1. Download the Docker image using this link: <https://mega.nz/folder/R2xTCLjC#dUrUkopEGuCT-0C4iejhVw>.

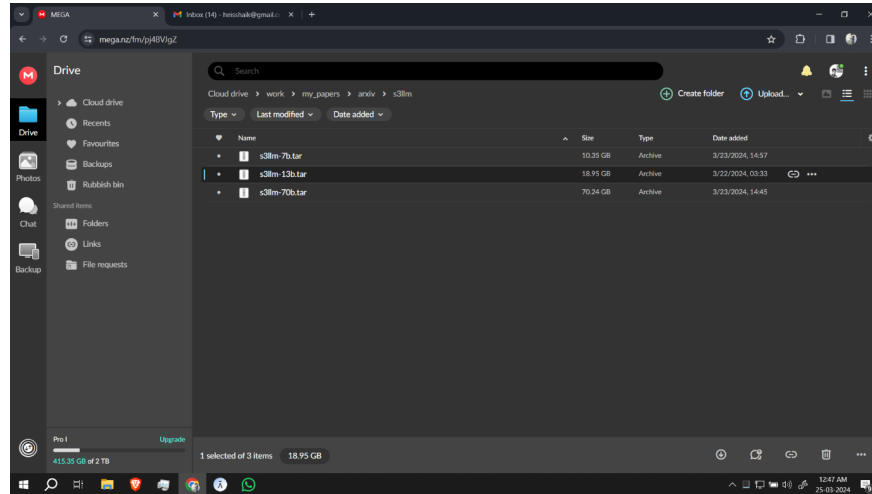


Figure 1: Download S3LLM docker images

2. Once you've downloaded the LLaMA-2 specific model Docker image, load it into Docker using the following command:

```
docker load < [file_name]
```

```
docker load < s3llm-13b.tar
```

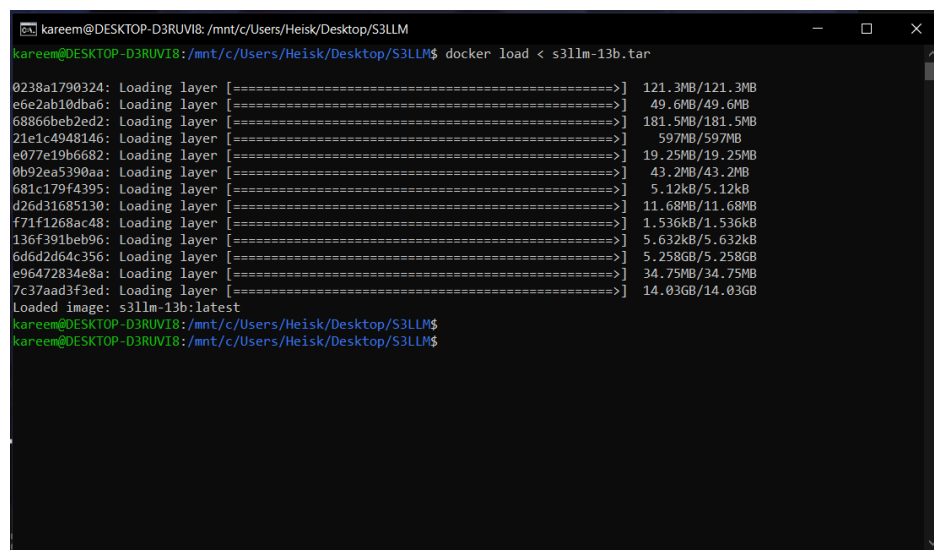
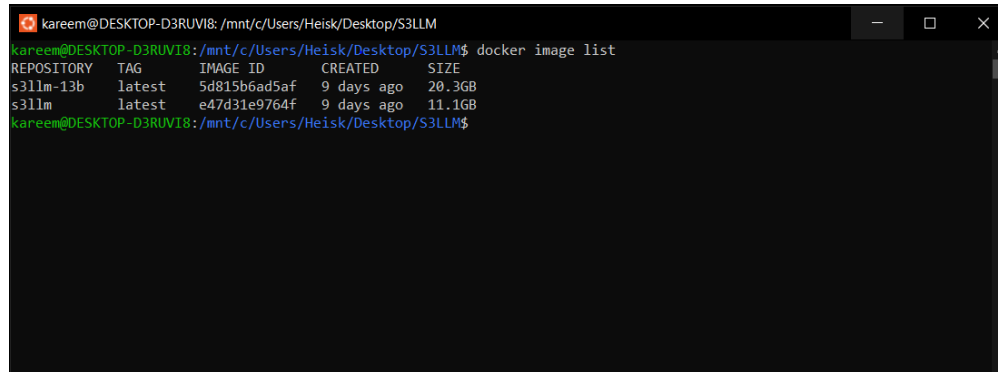


Figure 2: Load S3LLM Docker images

3. Verify if the image has been loaded properly by running:

```
docker image list
```



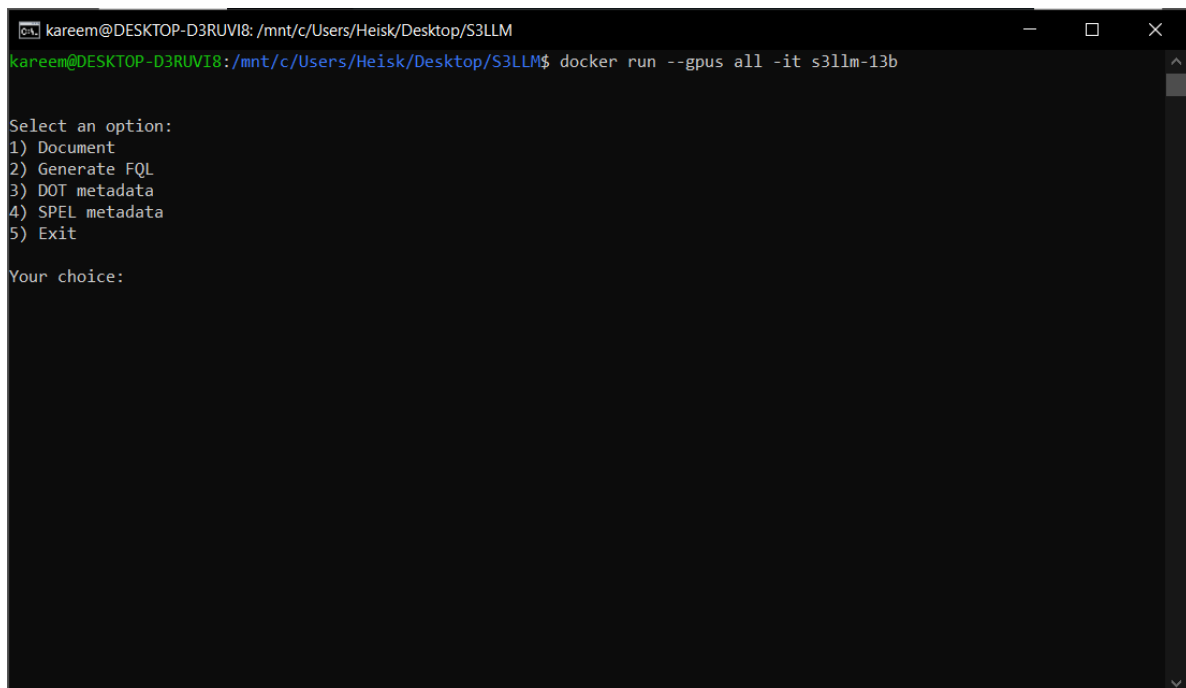
```
kareem@DESKTOP-D3RUVI8: /mnt/c/Users/Heisk/Desktop/S3LLM
kareem@DESKTOP-D3RUVI8:/mnt/c/Users/Heisk/Desktop/S3LLM$ docker image list
REPOSITORY TAG IMAGE ID CREATED SIZE
s3llm-13b latest 5d815b6ad5af 9 days ago 20.3GB
s3llm latest e47d31e9764f 9 days ago 11.1GB
kareem@DESKTOP-D3RUVI8:/mnt/c/Users/Heisk/Desktop/S3LLM$
```

Figure 3: GUI Version of Docker

4. To run the loaded image, use the following command:

```
docker run --gpus all -it [image_name]
```

```
docker run --gpus all -it s3llm-13b
```



```
kareem@DESKTOP-D3RUVI8: /mnt/c/Users/Heisk/Desktop/S3LLM
kareem@DESKTOP-D3RUVI8:/mnt/c/Users/Heisk/Desktop/S3LLM$ docker run --gpus all -it s3llm-13b

Select an option:
1) Document
2) Generate FQL
3) DOT metadata
4) SPEL metadata
5) Exit

Your choice:
```

Figure 4: Running Docker Image

5. In Figure 5, I have selected the first option to perform document analysis. Figure 6 displays the generated result.

```

kareem@DESKTOP-D3RUVI8: /mnt/c/Users/Heisk/Desktop/S3LLM
ama', 'llama.context_length': '4096', 'general.name': 'LLaMA v2', 'llama.embedding_length': '4096', 'llama.feed_forward_
length': '11008', 'llama.attention.layer_norm_rms_epsilon': '0.000001', 'llama.rope.dimension_count': '128', 'llama.atte
ntion.head_count': '32', 'tokenizer.ggml.bos_token_id': '1', 'llama.block_count': '32', 'llama.attention.head_count_kv':
'32', 'general.quantization_version': '2', 'tokenizer.ggml.model': 'llama', 'general.file_type': '17'}
Using fallback chat format: None

Enter your query: How is the Crank-Nicolson method applied in the context of this model?

```

Figure 5: Document Analysis

```

kareem@DESKTOP-D3RUVI8: /mnt/c/Users/Heisk/Desktop/S3LLM
llama_print_timings:      load time =  58667.15 ms
llama_print_timings:      sample time =   157.54 ms /  256 runs (  0.62 ms per token, 1625.03 tokens per second)
llama_print_timings: prompt eval time = 58666.69 ms / 288 tokens ( 203.70 ms per token,  4.91 tokens per second)
llama_print_timings:      eval time = 82347.97 ms / 255 runs ( 322.93 ms per token,  3.10 tokens per second)
llama_print_timings:    total time = 142160.84 ms / 543 tokens

AI: The Crank-Nicholson method is applied in this model by solving the system of equations obtained from Equation (9.44) using an implicit treatment of the time derivative. Specifically, at each time step  $t$ , the method updates the values of the variables in each layer using the following updates:

$$\begin{aligned} \frac{W_i^{n+1} - W_i^n}{\Delta t} &= \frac{1}{2} \left( \frac{F_i^n}{\Delta t} + \frac{F_i^{n-1}}{\Delta t} \right) \backslash, \backslash \frac{h_i^{n+1} - h_i^n}{\Delta t} &= \frac{1}{2} \left( \frac{L_i^n}{\Delta t} + \frac{L_i^{n-1}}{\Delta t} \right) \backslash, \backslash \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} &= \frac{1}{2} \left( \frac{R_i^n}{\Delta t} + \frac{R_i^{n-1}}{\Delta t} \right) \end{aligned}$$


Select an option:
1) Document
2) Generate FQL
3) DOT metadata
4) SPEL metadata
5) Exit

Your choice:

```

Figure 6: Result