# Llama.cpp Serving Demo with Openwebui

## Backend with Llama.cpp Serving

1. Build and install llama.cpp from source code
   1. install oneapi 2025.1 from this [link](https://www.intel.com/content/www/us/en/docs/oneapi/installation-guide-linux/2025-0/apt-005.html), and source vars.

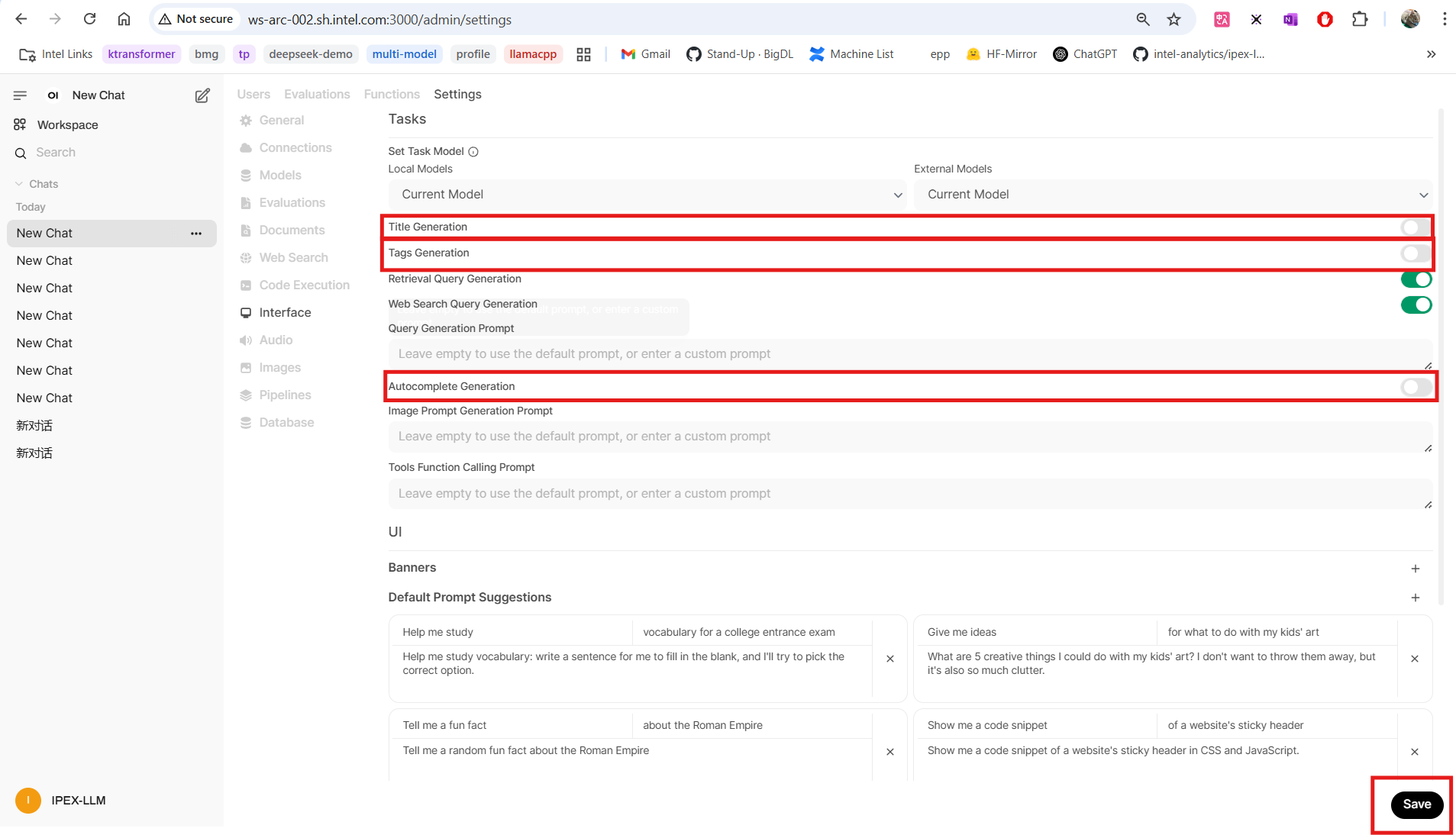
* source /home/intel/oneapi/setvars.sh
  1. build llama backend
* git clone https://github.com/intel-analytics/llm.cpp.git  
  cd llm.cpp/bigdl-core-xe/llama\_backend  
  mkdir build  
  cd build  
  cmake .. -DCMAKE\_C\_COMPILER=icx -DCMAKE\_CXX\_COMPILER=icpx  
  cmake --build . -j  
  sudo cp libllama\_bigdl\_core.a /usr/lib/
  1. build llama.cpp
* git clone https://github.com/intel-analytics/llama-cpp-bigdl.git  
   cd llama-cpp-bigdl  
   mkdir build  
   cd build  
   cmake .. -DGGML\_SYCL=ON -DCMAKE\_C\_COMPILER=icx -DCMAKE\_CXX\_COMPILER=icpx -DLLAMA\_BUILD\_TESTS=OFF -DGGML\_USE\_BIGDL=ON  
   cmake --build . --config Release -j -v  
   cd bin
  1. start DeepSeek-Q4\_K\_M using llama.cpp in the llama-cpp-bigdl/build/bin directory
     + t: thread, recomment to the physic core in the machine
     + c: ctx too large may lead to OOM of GPU, the biggest test is 10240
     + alias: the model name serving
     + port: the port of llamacpp serving
  + w9:
* export SYCL\_CACHE\_PERSISTENT=1  
   export SYCL\_PI\_LEVEL\_ZERO\_USE\_IMMEDIATE\_COMMANDLISTS=1  
   export IPEX\_LLM\_SCHED\_MAX\_COPIES=1  
   export IPEX\_LLM\_QUANTIZE\_KV\_CACHE=1 # to enable fp8 kv cache  
   /home/intel/junwang/llama-cpp-bigdl/build/bin/llama-server -m /home/intel/LLM/DeepSeek-R1-GGUF/DeepSeek-R1-Q4\_K\_M/DeepSeek-R1-Q4\_K\_M-00001-of-00009.gguf -t 60 -e -ngl 99 -c 10240 --temp 0 --no-context-shift -ot exps=CPU --no-mmap --host 0.0.0.0 --port 8001 --alias DeepSeek-R1-Q4\_K\_M
  + 云尖:
* export SYCL\_CACHE\_PERSISTENT=1  
   export SYCL\_PI\_LEVEL\_ZERO\_USE\_IMMEDIATE\_COMMANDLISTS=1  
   export IPEX\_LLM\_SCHED\_MAX\_COPIES=1  
   export IPEX\_LLM\_QUANTIZE\_KV\_CACHE=1 # to enable fp8 kv cache  
   /home/intel/junwang/llama-cpp-bigdl/build/bin/llama-server -m /home/intel/LLM/DeepSeek-R1-Q4\_K\_M/DeepSeek-R1-Q4\_K\_M-00001-of-00009.gguf -t 64 -e -ngl 99 -c 10240 --no-context-shift -ot exps=CPU --no-mmap --host 0.0.0.0 --port 8001 --alias DeepSeek-R1-Q4\_K\_M

## Frontend with Openwebui

### openwebui

1. Start docker using openwebui-520.sh to start openwebui version 5.20

bash openwebui-520.sh

1. Visit <http://localhost:3000> sign up or sign in
   * username: bigdl@intel.com
   * password: intel123
2. Since llamacpp’s serve has serious performance loss in concurrent situations, it is necessary to disable the three functions of openwebui that cause concurrency:
   1. visit <http://localhost:3000/admin/settings>
   2. click the Inferface button on left bar
   3. disable Title Generation, Tags Generation and Autocomplete Generation and click save to make the config work 
3. Add llamacpp serving api to Openwebui
   1. visit <http://localhost:3000/admin/settings>
   2. click the Connection button on left bar
   3. recommend to disable the Ollama API to reduce to useless request send to ollama serving address
   4. click + button on upper right, and url is the llama.cpp address http://localhost:8001/v1, api-key can be any string, click the sync button to check the api address is available, and save it. 