# Llama.cpp Serving Demo with Openwebui

## Performance settings

1. GPU Frequency setting:

sudo xpu-smi config -d 0 -t 0 --frequencyrange 2400,2400  
sudo xpu-smi config -d 1 -t 0 --frequencyrange 2400,2400

1. CPU Frequency setting: use lscpu to check the max frequency supported, and how many cores it has

sudo cpupower frequency-set -d 4.8GHz -u 4.8GHz

and cat /sys/devices/system/cpu/cpufreq/policy{0..120}/scaling\_cur\_freq to check the current frequency of cpu

1. Set machine to performance model, can use cpupower frequency-info to check

sudo cpupower frequency-set -g performance

## Backend with Llama.cpp Serving

### Step1: Get llama.cpp

**Using Release Version**

1. Find the download link on this page
2. Download and extract it

wget https://github.com/ipex-llm/ipex-llm/releases/download/v2.2.0/llama-cpp-ipex-llm-2.2.0-ubuntu-xeon.tgz  
tar -zxvf llama-cpp-ipex-llm-2.2.0-ubuntu-xeon.tgz

**Manual Build(optional)**

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

### Step2: start llama-server

Example llamacpp backend script is available llamacpp-backend.sh, change the LLAMA\_SERVER to the llamacpp path and MODEL to model path and start it using following command.

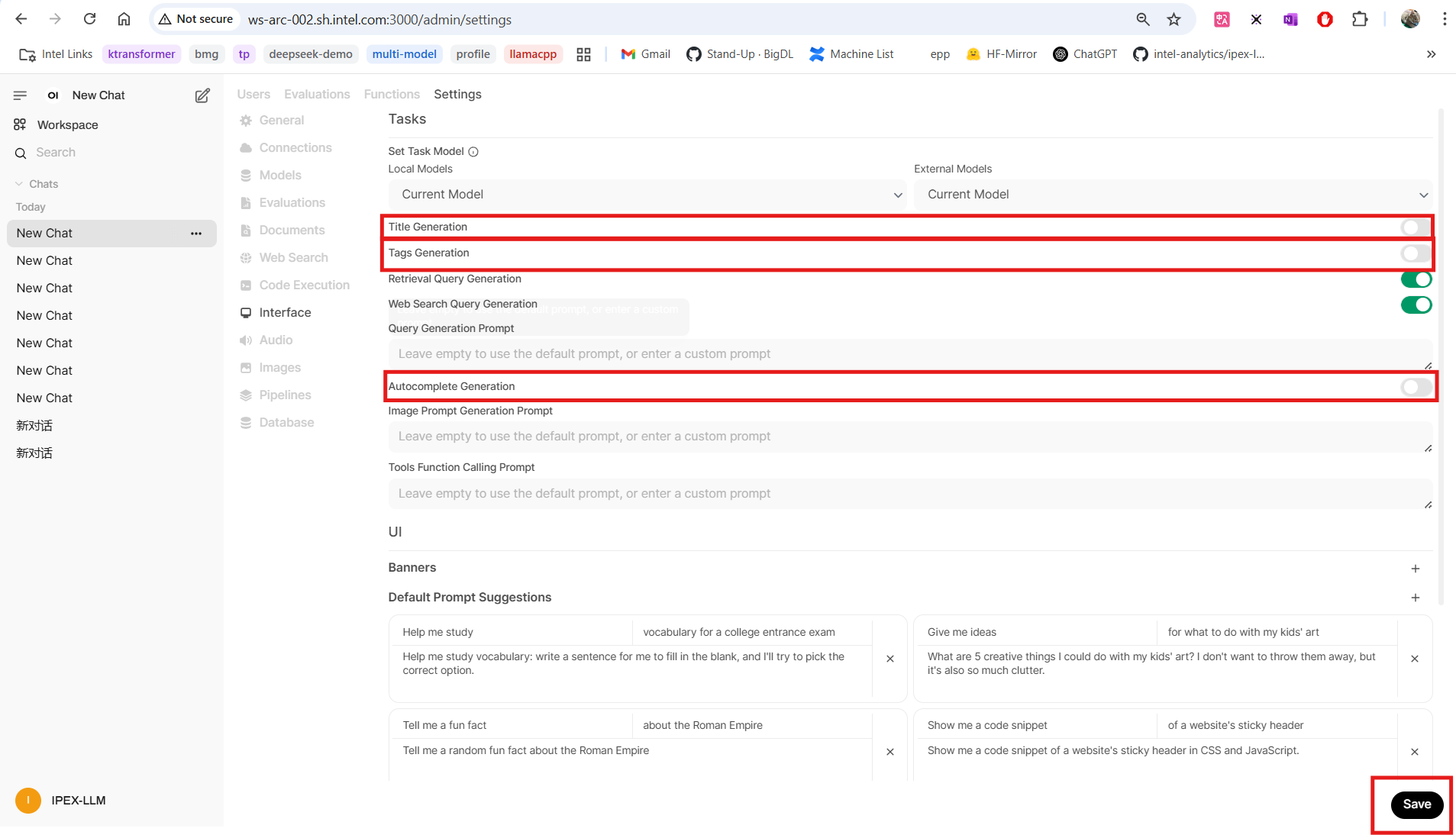
bash llamacpp-backend.sh

## 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. 