```
import requests
import json
from swarm_models.base_llm import BaseLLM



class llama3Hosted(BaseLLM):
    """
    A class representing a hosted version of the Llama3 model.

    Args:
        model (str): The name or path of the Llama3 model to use.
        temperature (float): The temperature parameter for generating responses.
        max_tokens (int): The maximum number of tokens in the generated response.
        system_prompt (str): The system prompt to use for generating responses.
        *args: Variable length argument list.
        **kwargs: Arbitrary keyword arguments.

    Attributes:
        model (str): The name or path of the Llama3 model.
        temperature (float): The temperature parameter for generating responses.
        max_tokens (int): The maximum number of tokens in the generated response.
        system_prompt (str): The system prompt for generating responses.

    Methods:
        run(task, *args, **kwargs): Generates a response for the given task.
```

```
"""


def __init__(
    self,
    model: str = "meta-llama/Meta-Llama-3-8B-Instruct",
    temperature: float = 0.8,
    max_tokens: int = 4000,
    system_prompt: str = "You are a helpful assistant.",
    base_url: str = "http://34.204.8.31:30001/v1/chat/completions",
    *args,
    **kwargs,
):
    super().__init__(*args, **kwargs)
    self.model = model
    self.temperature = temperature
    self.max_tokens = max_tokens
    self.system_prompt = system_prompt
    self.base_url = base_url


def run(self, task: str, *args, **kwargs) -> str:
    """
    Generates a response for the given task.


    Args:
        task (str): The user's task or input.
```

```
Returns:
    str: The generated response from the Llama3 model.

"""

payload = json.dumps(
    {
        "model": self.model,
        "messages": [
            {"role": "system", "content": self.system_prompt},
            {"role": "user", "content": task},
        ],
        "stop_token_ids": [128009, 128001],
        "temperature": self.temperature,
        "max_tokens": self.max_tokens,
    }
)

headers = {"Content-Type": "application/json"}

response = requests.request(
    "POST", self.base_url, headers=headers, data=payload
)

response_json = response.json()
assistant_message = response_json["choices"][0]["message"][
    "content"
]

return assistant_message
```