Workflow for implementing and training custom code in a cloud setting (LoRA in this case just as an example)

ALWAYS READ THIS FOR MORE DETAILS :
https://cloud.google.com/vertex-ai/docs/training/create-custom-job

Step 1: Prepare Training Code with BERT and LoRA (test peft implementation locally before packaging)

[Location : Local Machine] - Load the pre-trained BERT model (Option 1 : Hugging Face Transformers or Option2 : TensorFlow/Keras).

ex: `AutoModelForSequenceClassification` or `TFAutoModel`

- Add LoRA layers to the BERT model

Option1 : `peft` library for PyTorch : https://github.com/huggingface/peft

Option2 : custom layer injection for TensorFlow

- Define the training loop with dataset and task-specific loss function(should define the loss function)

- Save notes in `*requirements.txt*` and test locally

Step 2: Autopackage and Push Container (Cloud Shell)
[Location : Google Cloud Shell]
- Use `gcloud` CLI to:
  - Package code into a Docker container
  - Push container to Artifact Registry

Step 3: Configure Custom Training Job (Cloud Shell)
[Location : Google Cloud Shell]
  - Machine type (ex. n1-standard-4)
  - GPU/TPU (probably won't work but we can try)
- Submit training job : use `gcloud ai custom-jobs create`

Step 4: Training (Vertex AI VM)
[Location : Vertex AI VM]
- Vertex AI runs on our containerized training job
- LoRA layers are fine-tuned

Step 5: Evaluate and Save Model (Vertex AI VM)
[Location : Vertex AI VM]
- Save the fine-tuned model to Google Cloud Storage
- Evaluate bias metrics (should define)

Step 6: Deploy (Google Cloud or External BERT)

Example github repository that specifically leverages BERT for debiasing :
https://github.com/IMPLabUniPr/BERT-for-ABSA