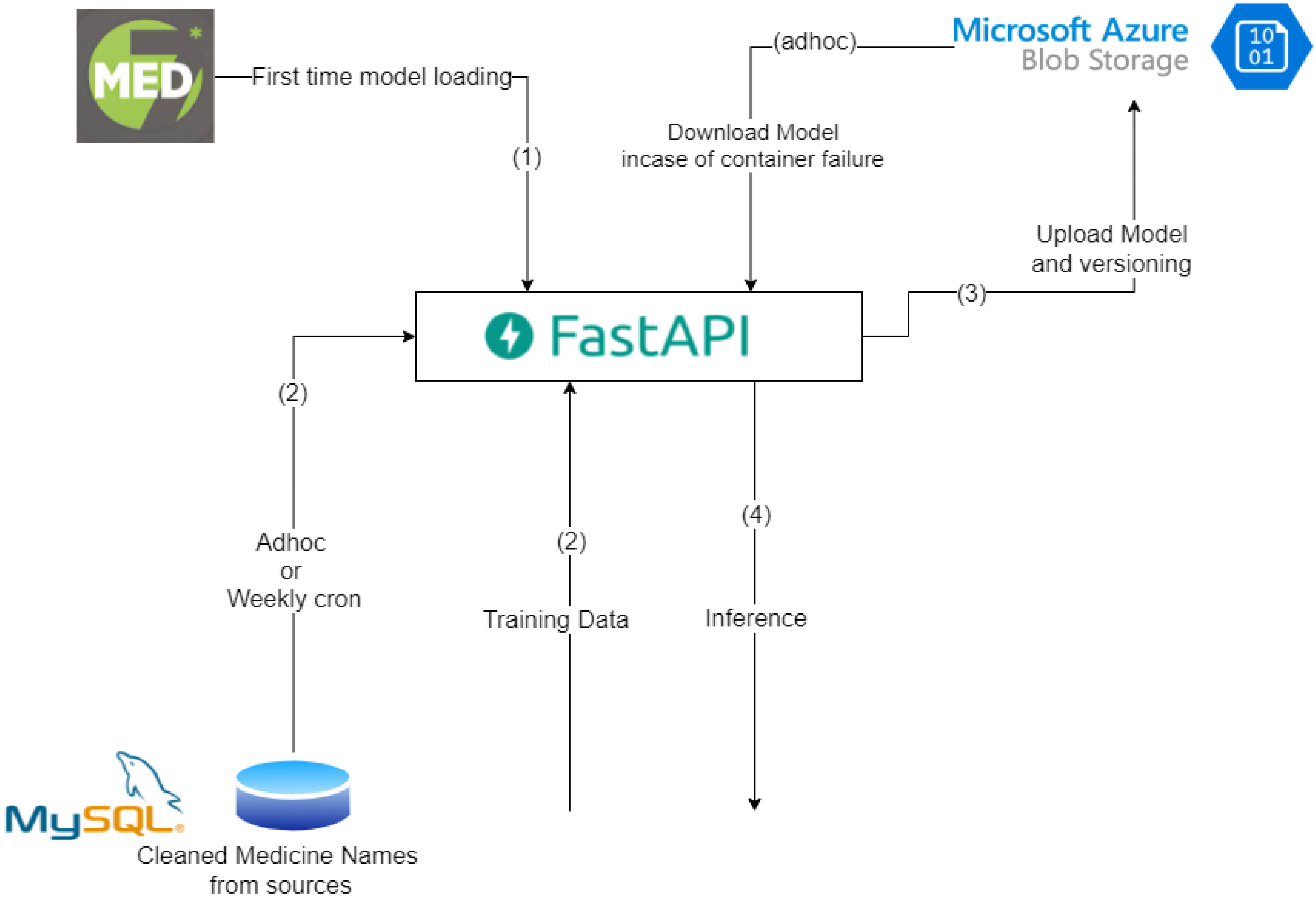
MED
First time model loading
(1)
(adhoc)
Microsoft Azure
Blob Storage
10
01
Download Model
incase of container failure
Upload Model
and versioning
(3)
FastAPI
(2)
Adhoc
or
Weekly cron
(2)
Training Data
(4)
Inference
MySQL
Cleaned Medicine Names
from sources

Repo Link:
https://dev.azure.com/BFHL/InsightsRx/_git/med7 - Voice prescription MED7 API

Note - It is highly recommended that even on vm this should be deployed as a docker container to avoid conflicts with future deployments and packages.

Execution Flow:

TRAINING:

1) Currently Training for new medicines is supported. This api has a training endpoint which accepts a txt file of medicine names separated on new lines. Please ensure the medicine names are cleaned before processing (no special characters like : ; and commas in name)
2) The deployment will use the base model of med7 only for the first time. In case of VM failure the recently saved checkpoint model from azure blob will be pulled to resume training from the last checkpoint.
3) Pass a txt file to training endpoint of api
4) For fine tuning the spacy model we require already trained data as well in the pipeline to avoid Catastrophic forgetting problem. After the txt data is processed , the pipeline automatically adds 10% of txt file data count. For Eg. If you want to train 100 medicines in a txt file , the pipeline will automatically add 10 more medicines which are already trained on the model. This old medicine file is PRESCRIPTIONS.txt . Access to whole dataset requires registration on scholar site (please replace this small sample file with whole dataset for better results)
5) The data received is raw and has to be gold annotated to be trained on spacy model. Gold annotation is (text, {"entities": [(0, len(text), "DRUG")]}) basically a tuple of text and list of tuples which contains indexes to be annotated with the given tag. Check spacy site for more info.
6) Training the data on model -
   -Optimizer is used from the pretrained model
   -Data is processed in batch form
   -The batch data is randomly shuffled for better results
   -Currently trains for 10 epochs
   -Data has to be passed to model in doc form
   -Training logs and losses are recorded in traininglogs.txt
   -drop parameter has been set 0.1 (According to model's default config)
7) Saving model to disk. This is going to be used for versioning and packaging for an installable spacy library. Local version.txt file stores the iteration for fine tuning and updates everytime the model is trained. When the model is saved to disk , we run a terminal command from spacy package which converts the files to a tar.gz which can be used as a pip install library. Note: Name for packaged library is always same without any explicit version mentioned. We will transfer the versioned files to azure cdn blob for model version management and the executable tar.gz library will be replaced without any name change on azure cdn blob.
8) Upload to Azure CDN Blob. ebhmcontainer has been specially enabled public cdn access so that inference container can pull this library and install for inference. Dist folder in blob container will have tar.gz installable always with the same name. Models folder in blob container will have multiple versions of trained checkpoints for model to be used for versioning storage.
9) Local files will be deleted from container to free space and clean up resources
10) Training has completed

INFERENCE API:

1) Current model will be the trained model so a reload is not required.
2) Inference is a fastapi endpoint which serves with a single inference endpoint
3) This endpoint accepts a string and outputs

Weekly Automated training or Ad Hoc automated training:

1) From _ database _ table will pick up new deduped untrained data.
2) This will be weekly cron job which will train and release a new model with fresh untrained data
3) _ table has medicine name(text) - medicine name text , trained(1 or null) - flag if model has trained on this data , model version(text) - this will denote the model version for which data has been trained.