**NAMED ENTITY RECOGNITION**

**Initial steps**

1. Create a local repo in your system
2. Create README.md , .gitignore , LICENSE
3. Create the template.py
4. Create the environment and activate it
5. Created the folder structure

**Gcloud**

1. Create an acc in gcloud for data ingestion
2. Then install gcloud CLI
3. Run the command ”gcloud init”
4. Install the requirements.txt
5. And install the torch

**Logger , Exception , constants and utils**

1. Ner🡪 logger
2. Ner🡪exception
3. Ner🡪 constants
4. Ner 🡪 utils

**Configuration setup with GCloud**

1. Update Ner🡪 configuration🡪 gcloud.py
2. Project 🡪NER-BERT🡪create project
3. Gcloud 🡪 Buckets🡪 create bucket🡪 name🡪 create 🡪 confirm

Now we have to upload the data

1. Buckets🡪 upload file🡪 upload

Now configure the account

1. Cloud init🡪 new configuration 2 🡪 name🡪mailid 🡪 authenticate
2. Created a test folder and import the data from gcloud

**Notebook experiment**

1. Do the experiment in colab

**Workflow**

1. **Constants**
2. **Config\_entity**
3. **Artifact\_entity**
4. **Components**
5. **Pipeline**
6. **End point**

**MODULAR CODING**

**Data Ingestion**

1. Update ner🡪 constants
2. Ner🡪 entity🡪 config\_entity
3. Ner 🡪 entity 🡪 artifact\_entity
4. Ner 🡪 componets 🡪 data ingestion.py
5. Ner 🡪 pipeline🡪 train\_pipeline
6. End point

**Data Transformation**

1. Update ner🡪 constants
2. Ner🡪 entity🡪 config\_entity
3. Ner 🡪 entity 🡪 artifact\_entity
4. Ner 🡪 componets 🡪 data transformation.py
5. Ner 🡪 pipeline🡪 train\_pipeline
6. End point

**Model Trainer**

1. Update ner🡪 constants
2. Ner🡪 entity🡪 config\_entity
3. Ner 🡪 entity 🡪 artifact\_entity
4. Ner 🡪 componets 🡪 model trainer.py
5. Create a folder model 🡪 create a constructor file 🡪create another file : bert.py 🡪create the model architecture
6. Ner 🡪 pipeline🡪 train\_pipeline
7. End point

**Model Evaluation**

1. Update ner🡪 constants
2. Ner🡪 entity🡪 config\_entity
3. Ner 🡪 entity 🡪 artifact\_entity
4. Ner 🡪 componets 🡪 model evaluation.py
5. Ner 🡪 pipeline🡪 train\_pipeline
6. End point

**Model Pusher**

1. Ner🡪 entity🡪 config\_entity
2. Ner 🡪 entity 🡪 artifact\_entity
3. Ner 🡪 componets 🡪 model pusher.py
4. Ner 🡪 pipeline🡪 train\_pipeline
5. End point

**Prediction Pipeline**

1. Ner🡪entity🡪config\_entity
2. Ner🡪pipeline🡪prediction\_pipeline.py

**User App**

1. Update app.py (FastAPI)

**Deployment**

1. Create a .circleci folder
2. And inside that create config.yaml
3. Update Dockerfile
4. Update .dockerignore
5. Create a folder scripts
6. Inside scripts 🡪 VM-machine-setup.sh
7. **GCP account**
8. Artifacts registry🡪enable API 🡪create repository
9. Name: <---------> , mode:standard , location : region :asia south Mumbai 🡪 create
10. Ner-bert 🡪copy the url
11. Change the config.yaml in .circleci folder
12. Copy the url in lines 42,50,72,76
13. **Circleci**
14. Create a circleci account
15. Home page🡪Organization settings 🡪 self hosted runners 🡪 agree the terms
16. Projects🡪new project🡪 github🡪 project name: <------->,select the repo 🡪create project
17. Project🡪 <------> 🡪project settings 🡪 environment variable 🡪 add environment variable
18. GCLOUD\_SERVICE\_KEY --> service account , GOOGLE\_COMPUTE\_ZONE = asia-south1 , GOOGLE\_PROJECT\_ID
19. Email:--------------------------------------------------------------------
20. Ner-bert-project
21. Gcloud 🡪 service accounts 🡪 IAM 🡪 create service 🡪 keys🡪 download the value in the json file 🡪 copy : GCLOUD\_SERVICE\_KEY
22. Add env variable
23. GOOGLE\_COMPUTE\_ZONE = asia-south1
24. Add env variable
25. Gcloud home page🡪 copy project id 🡪 GOOGLE\_PROJECT\_ID : ner-bert-423605

**Now create a virtual instance**

1. GCPCloud 🡪vm instance 🡪 create instance 🡪enable
2. Create Instance 🡪 instance name: ner-bert 🡪location: asia south Mumbai 🡪 preset🡪standard 🡪 atleast 8GB
3. Boot disk : ubuntu 🡪 ubuntu 20.04 LTS 🡪default 🡪 atleast 50 GB🡪 select
4. Select the service acc 🡪 allow HTTP , HTTPS 🡪 create
5. Instance 🡪 SSH 🡪 open in browser 🡪 run the commands in vm-machine-setup

**Setup the gCloud**

1. Circleci 🡪selfhosted runner 🡪create selfhosted runner 🡪name:organization name (datascience) , resource class label: nerbert 🡪 create
2. Copy the token in vm-machine-setup.sh
3. Copy the selfhosted runner name in vscode line :54 organisation name/resource class label
4. Copy the configuration file from the config.yaml and run in the terminal
5. Ctrl + s 🡪 ctrl + x
6. After all the setup 🡪 pause and restart the vm instances
7. Push the change to github
8. And go to circleci pipeline and cicd deployment will start ...