1. GRIDPOT SETUP

This appendix summarizes the steps required to install GridPot. We changed our virtual-machine network setting to use NAT during this setup.

1. Installing dependencies (from the command line)

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get install autoconf (this includes automake)

$ sudo apt-get install libtool subversion python-dev mysql-server

$ sudo apt-get install libmysqlclient-dev libmysqld-dev

$ sudo apt-get install libxerces-c-dev python-mysqldb python-pip

$ sudo apt-get install libcurl3 libcurl4-openssl-dev libcurl4- gnutls-dev

$ pip install –U sphinx $ sudo apt-get install python3-sphinx libxml2-dev libxslt1-dev

$ sudo pip install lxml gevent python-dateutil mixbox

$ sudo pip install pyasn1 pycryptodomex pysmi

$ sudo apt-get install doxygen

$ sudo apt-get install libcppunit-dev libcppunit-doc

$ sudo apt-get install libncurses5-dev libncursesw5-dev

1. Pulling GridPot from GitHub

$ sudo apt install git

$ git clone <https://github.com/sk4ld/gridpot.git>

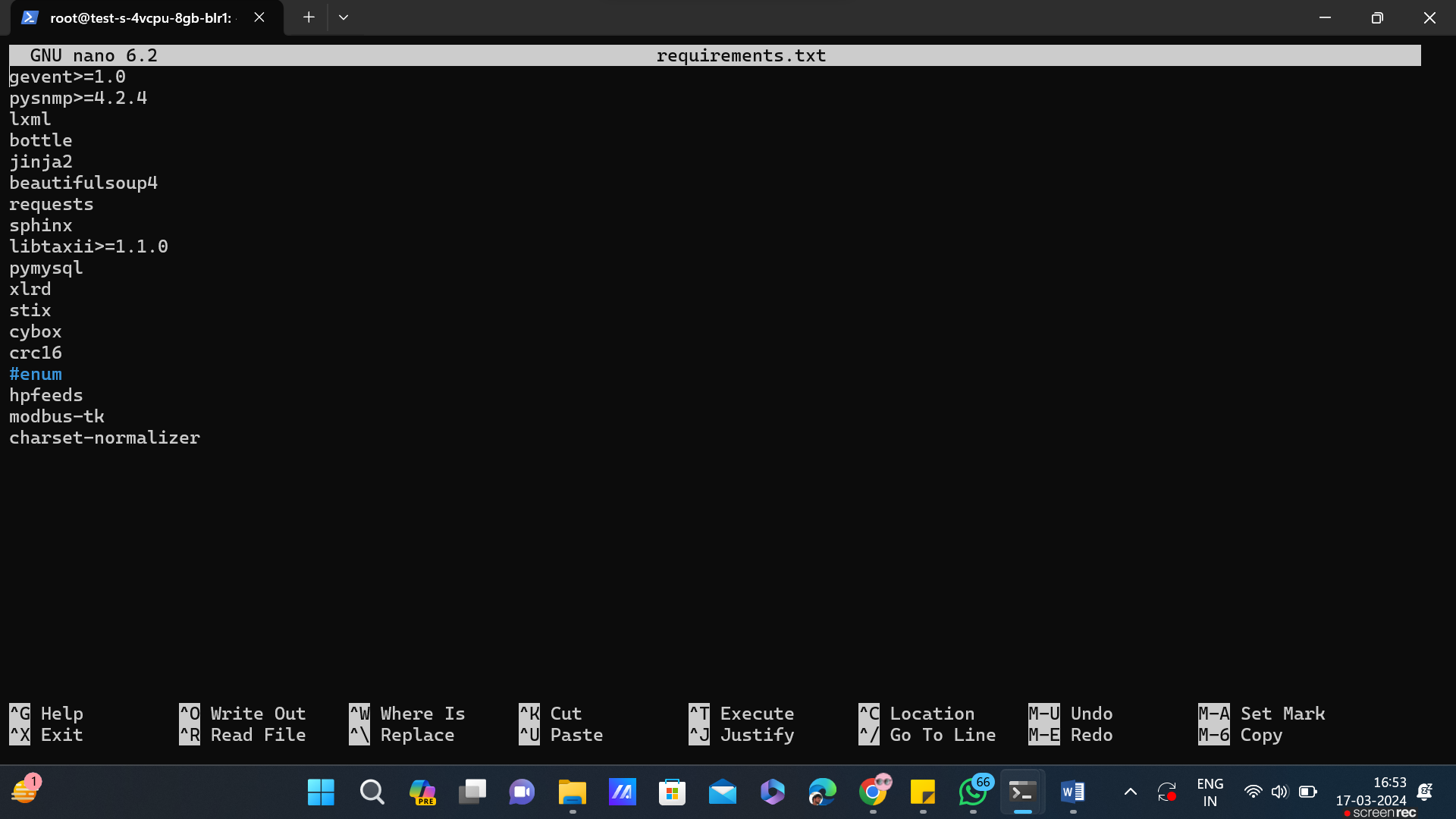
Move to /gridpot rm –rf gridlabd and then

git clone https://github.com/gridlab-d/gridlab-d.git

3. Setup Conpot

$ cd gridpot/conpot/

Use these dependencies in requirements.txt file



Now, replace the setup.py with new setup.py file from <https://github.com/mushorg/conpot/blob/master/setup.py>

$ conpot/ sudo make clean (if rebuilding) $ conpot/ sudo python setup.py install 46

4. Setup GridLAB-D

$ cd ../gridlab-d

$ sudo git submodule update --init

$ mkdir cmake-build

$ cd cmake-build

$ sudo cmake ..

$ sudo make

$ sudo make install$ make $ sudo make install 5. Setup libiec61850 $ cd ../libiec61850 $ make $ sudo make INSTALL\_PREFIX=/usr/local install

ow it's time to run the gridpot and follow the command from below

$ cd gridpot/gridlab-d/models (Path)

$ gridlabd -D run\_realtime=1 --server --debug --verbose IEEE\_13\_Node\_With\_Houses.glm 2>&1 | tee HousesOutput.txt

now,

$ cd gridpot/conpot/conpot/templates (Path)

$ sudo conpot -t gridpot -v

So here the Gridpot is a bit troublesome, as we know that gridpot has its last updated nearly 8 years ago. So the framework is old, but here's the logic to utilise the framework in our environment.

the last command may give us some errors, at paths of binary/executable files which were installend after executing setup.py file for conpot host installation.

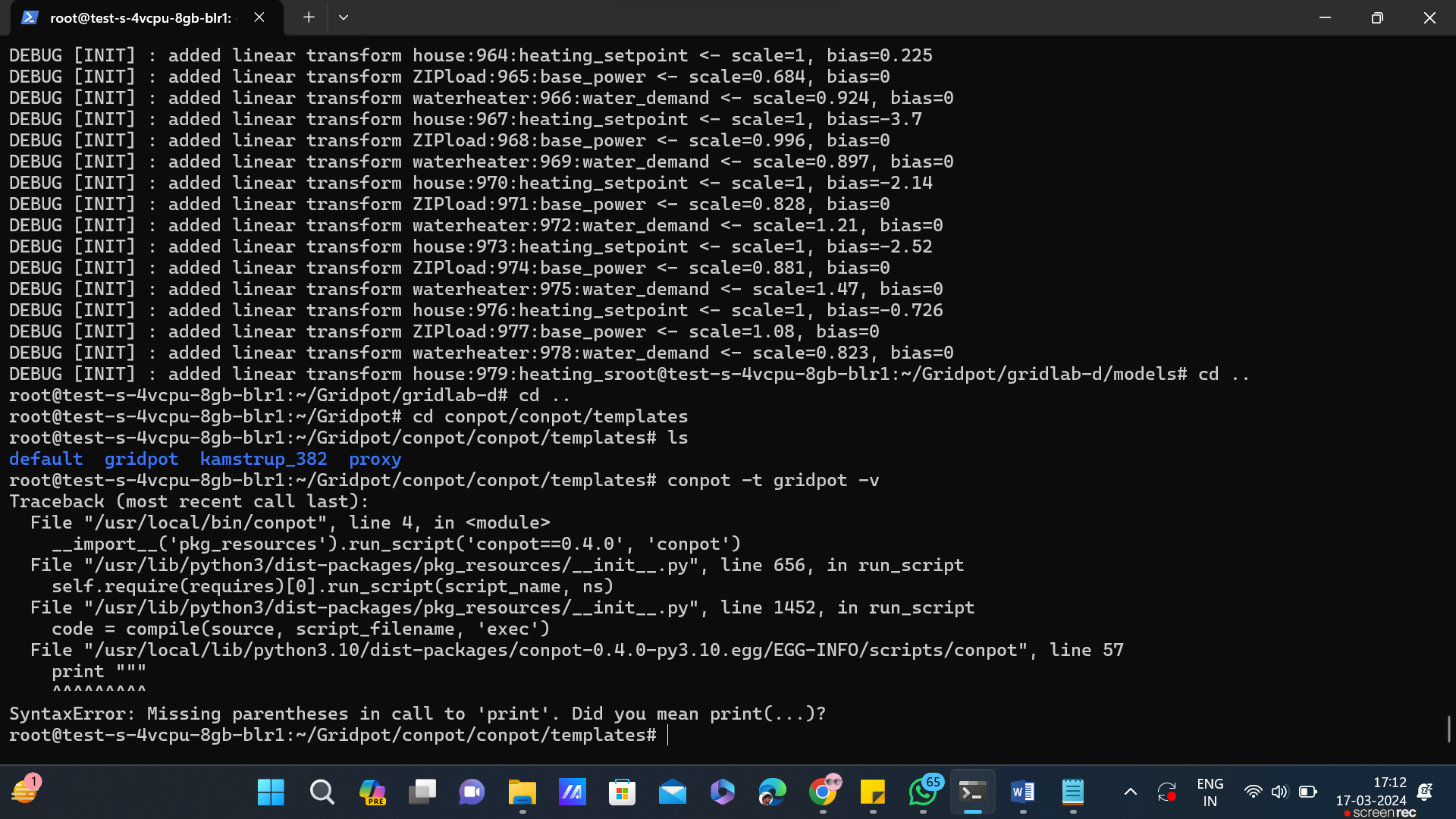
Just follow the error path using cd or directly access the files using any editor, i used nano(user-friendly)

let me guide you that there are threads of errors

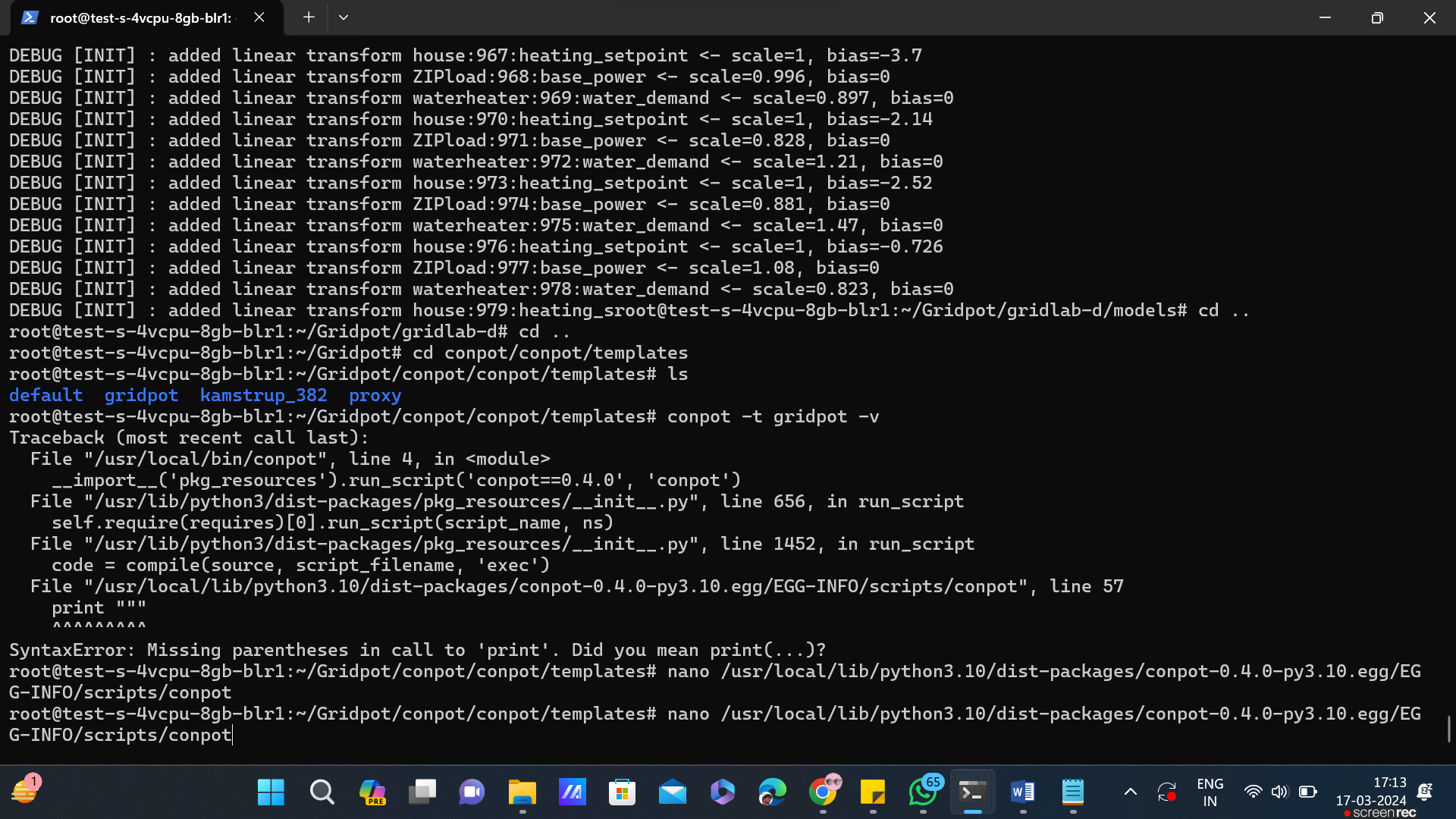
1. module not found, for that use correct path if it is a custom python package and when it is a pypi package then replace it with latest version name

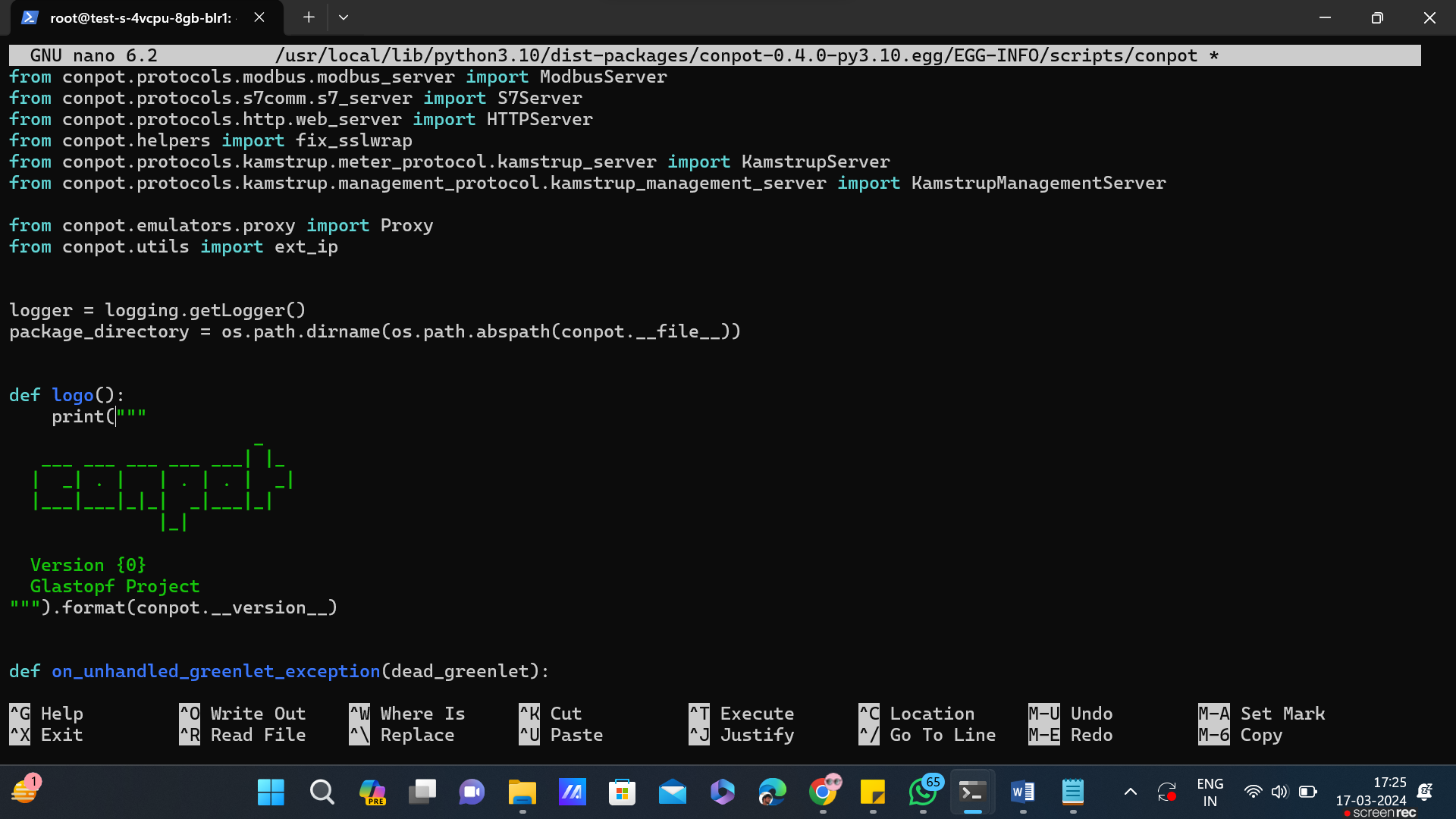
for example there is this error related to ConfigParser module not found in line import ConfigParser, replace python code with from configparser import ConfigParser. follow likewise for similar errors.

2. syntax errors :- the code would be in python2.x just udpate with python3.x syntax (easy the debug such type of errors).



Solution :-



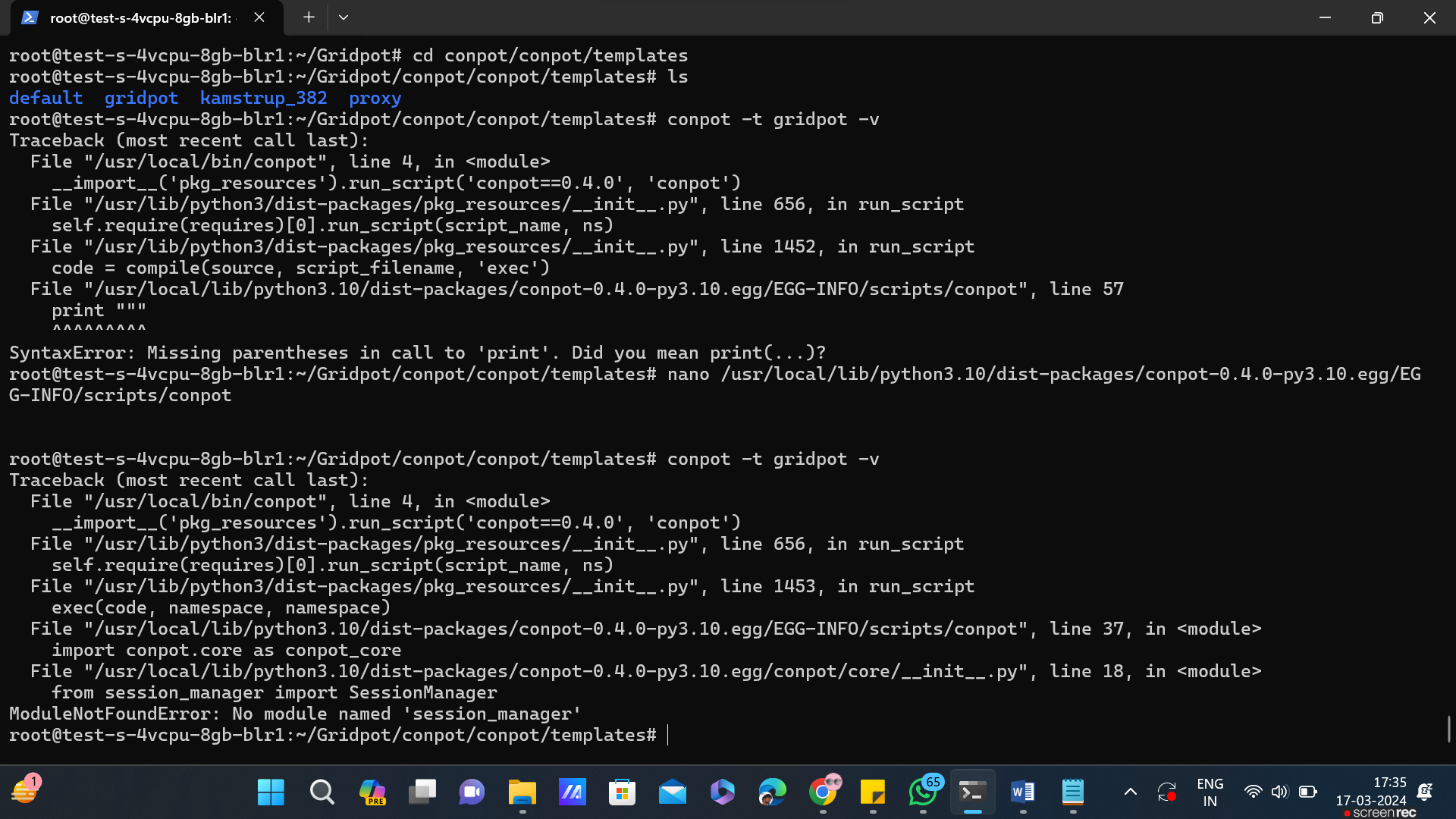


Go to the location doing nano and solve the error as shown in above screenshot

Then re run the command

conpot –t gridpot –v

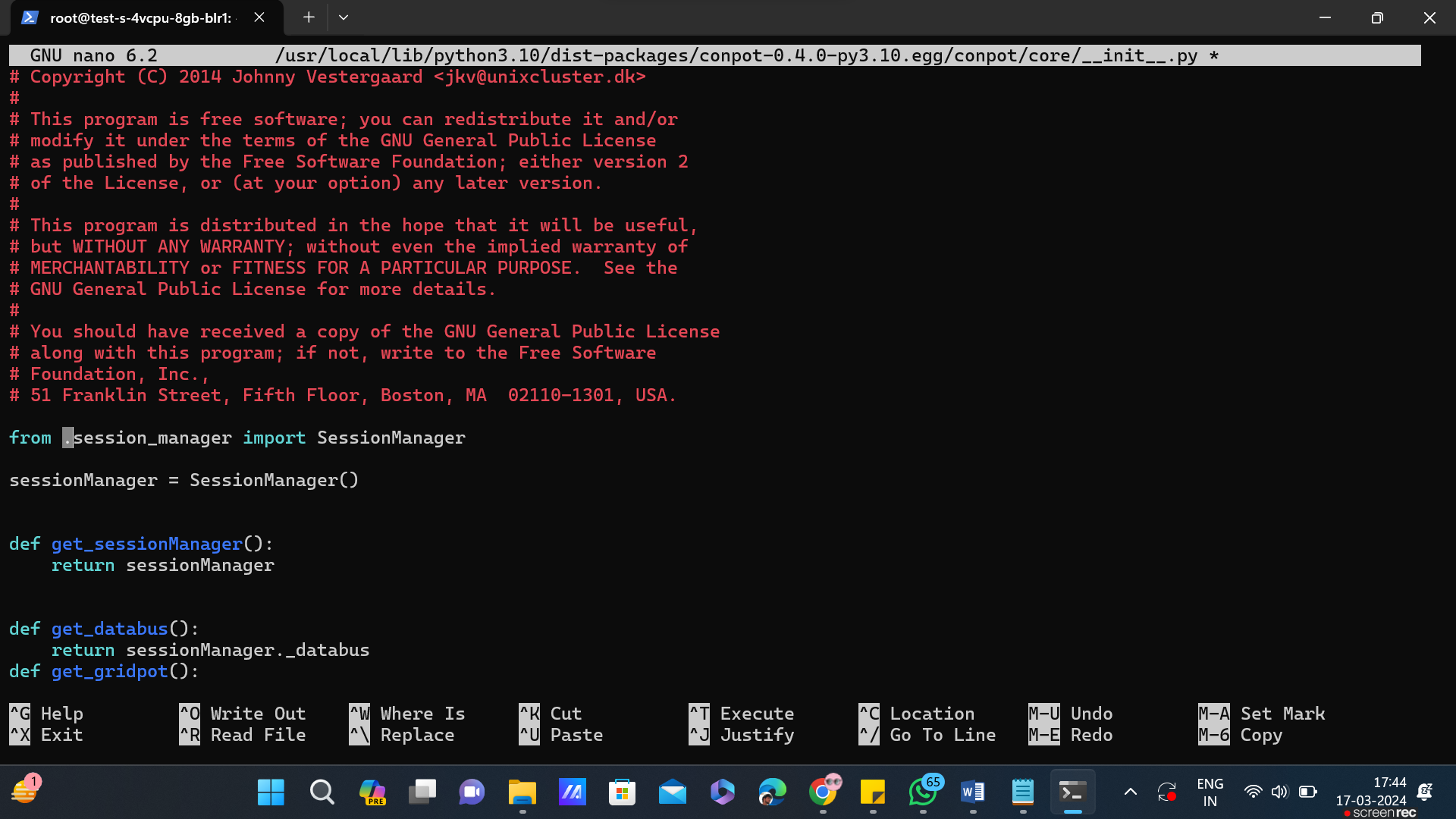
another like this

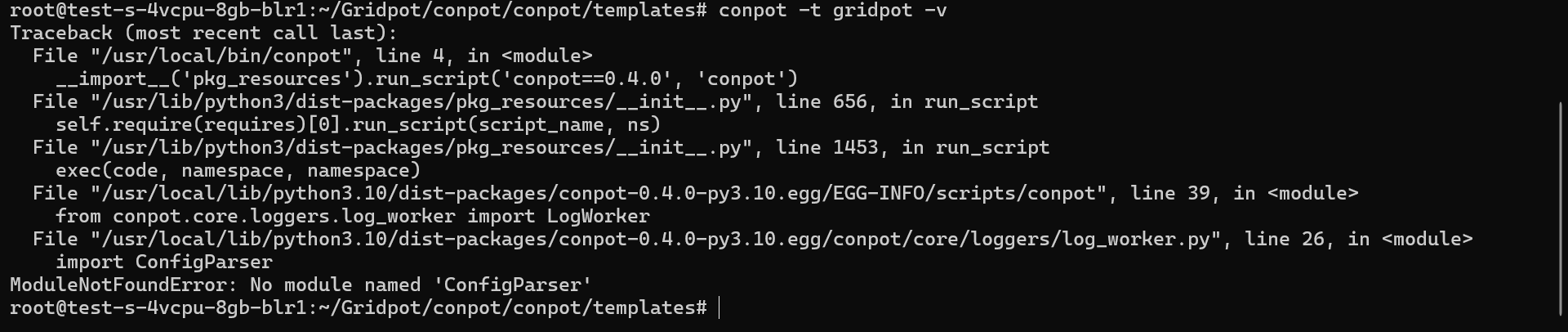


For above error solve it by doing to the error location

nano /usr/local/lib/python3.10/dist-packages/conpot-0.4.0-py3.10.egg/conpot/core/\_\_init\_\_.py

then solve the error like given below in the screenshot





Solution is same go to the error doing nano and correct

It in the line of error as from configparser import ConfigParser

Now, re run the command

And keep solving the errors like this

First going to the place of error, then line of error and then correct the syntactical error.

Re run the command conpot –t gridpot –v

And keep solving such errors

At last when the error are solved you can deploy thr Gridpot using below docker-compose.yml file , before running docker-compose.yml run the gridpot in the background by using multiple choices.

Like nohup, 2&1 command, etc.

version: '3.1'

services:

agent:

build:

context: .

dockerfile: ./agent/agent\_dockerfile

container\_name: HoneyPot\_Agent

restart: always

networks:

- elk

ports:

- '9090:9090'

elasticsearch:

image: docker.elastic.co/elasticsearch/elasticsearch:7.7.0

networks:

- elk

environment:

cluster.name: elk

network.host: 0.0.0.0

discovery.type: single-node

ES\_JAVA\_OPTS: "-Xms256m -Xmx256m"

filebeat:

image: docker.elastic.co/beats/filebeat:7.7.0

command: ["--strict.perms=false", "-c", "/usr/share/filebeat/filebeat.yml"]

volumes:

- ./suricata/logs:/var/log/suricata:ro

- ./filebeat/filebeat.yml:/usr/share/filebeat/filebeat.yml:ro

networks:

- elk

environment:

LS\_JAVA\_OPTS: "-Xms256m -Xmx256m"

restart: always

depends\_on:

#- kibana

- suricata

#kibana:

# image: docker.elastic.co/kibana/kibana:7.7.0

#networks:

# - elk

#environment:

# server.name: "kibana"

#server.host: "0"

#elasticsearch.hosts: '[ "http://elasticsearch:9200" ]'

#ports:

# - 5601:5601

#depends\_on:

# - elasticsearch

suricata:

image: jasonish/suricata:latest

command: -i eth0

network\_mode: host

cap\_add:

- NET\_ADMIN

- SYS\_NICE

volumes:

- ./suricata/logs:/var/log/suricata

- ./suricata/rules:/var/lib/suricata

- ./suricata/config:/etc/suricata

depends\_on:

- elasticsearch

networks:

frontend:

internal: false

driver: bridge

driver\_opts:

com.docker.network.bridge.name: hp\_core

backend:

internal: true

elk: