



The science behind the report:

Ingesting data for use with a large language model for AI: Latest-generation Dell PowerEdge servers powered by 5th Generation AMD EPYC processors offer a range of strong options

This document describes what we tested, how we tested, and what we found. To learn how these facts translate into real-world benefits, read the report [Ingesting data for use with a large language model for AI: Latest-generation Dell PowerEdge servers powered by 5th Generation AMD EPYC processors offer a range of strong options](#).

We concluded our hands-on testing on July 24, 2025. During testing, we determined the appropriate hardware and software configurations and applied updates as they became available. The results in this report reflect configurations that we finalized on July 11, 2025 or earlier. Unavoidably, these configurations may not represent the latest versions available when this report appears.

Our results

To learn more about how we have calculated the wins in this report, go to <http://facts.pt/calculating-and-highlighting-wins>. Unless we state otherwise, we have followed the rules and principles we outline in that document.

Table 1: Results of our testing.

	Processor(s)	float32 precision Sentences per second	bfloat16 precision Sentences per second
Dell PowerEdge R7715			
Configuration 1	1x 5 th Generation AMD EPYC 9575F processor	345.2	1,046.3
Configuration 2	1x 5 th Generation AMD EPYC 9745 processor	461.1	1,486.3
Configuration 3	1x 5 th Generation AMD EPYC 9845 processor	510.4	1,660.4
Dell PowerEdge R7725			
Configuration 1	2x 5 th Generation AMD EPYC 9575F processors	656.3	1,965.5
Configuration 2	2x 5 th Generation AMD EPYC 9965 processors	1,094.8	3,907.4

System configuration information

Table 2: Detailed information on the systems we tested.

System configuration information	Dell PowerEdge R7715 server	Dell PowerEdge R7725 server
BIOS name and version	Dell 1.2.4	Dell 1.2.4
Non-default BIOS settings	N/A	N/A
Operating system name and version/build number	Ubuntu 25.04 Kernel version 6.14.0-23-generic	Ubuntu 25.04 Kernel version 6.14.0-23-generic
Date of last OS updates/patches applied	7/10/25	7/10/25
Power management policy	Performance	Performance
Processor configuration 1		
Number of processors	1	2
Vendor and model	AMD EPYC™ 9575F	AMD EPYC™ 9575F
Core count (per processor)	64	64
Core frequency (GHz)	3.3	3.3
Max turbo frequency (GHz)	5.0	5.0
All-cores turbo frequency (GHz)	4.5	4.5
L3 cache per processor (MB)	256	256
Stepping	1	1
Processor configuration 2		
Number of processors	1	2
Vendor and model	AMD EPYC™ 9745	AMD EPYC™ 9965
Core count (per processor)	128	192
Core frequency (GHz)	2.4	2.25
Max turbo frequency (GHz)	3.7	3.7
All-cores turbo frequency (GHz)	3.45	3.35
L3 cache per processor (MB)	256	384
Stepping	0	0
Processor configuration 3		
Number of processors	1	N/A
Vendor and model	AMD EPYC™ 9845	N/A
Core count (per processor)	160	N/A
Core frequency (GHz)	2.1	N/A
Max turbo frequency (GHz)	3.7	N/A
All cores turbo frequency (GHz)	3.25	N/A
L3 Cache per processor (MB)	320	N/A
Stepping	0	N/A

System configuration information	Dell PowerEdge R7715 server	Dell PowerEdge R7725 server
Memory module(s)		
Total memory in system (GB)	768	1,536
Number of memory modules	12	24
Vendor and model	Micron® MTC40F2046S1RC64BD2	Micron® MTC40F2046S1RC64BD2
Size (GB)	64	64
Type	PC5-51200	PC5-51200
Speed (MHz)	6,400	6,400
Speed running in the server (MHz)	5,200	6,400
Local storage (type A)		
Number of drives	8	8
Drive vendor and model	Dell NVMe® PM1743 RI E3.S 3.84TB	Dell NVMe® PM1743 RI E3.S 3.84TB
Drive size (GB)	3,840	3,840
Drive information (running speed, interface, type)	PCIe gen5x2, NVMe, SSD	PCIe gen5x4, NVMe, SSD
Purpose	OS, application	OS, application
Network adapter		
Vendor and model	iDRAC USB Ethernet	iDRAC USB Ethernet
Number and type of ports	1 x 1GbE	1 x 1GbE
Driver version	cdc_ncm 6.14.0-23-generic	cdc_ncm 6.14.0-23-generic
Cooling fans		
Vendor and model	Dell Platinum	Dell Platinum
Number of cooling fans	6	6
Power supplies		
Vendor and model	Dell 0GRD17A01	Dell 0GRD17A00
Number of power supplies	2	2
Wattage of each (W)	1,500	1,500

How we tested

Software versions

- **OS:** Ubuntu Server 25.04
- **Linux kernel:** 6.14.0-23-generic
- **Docker:** docker-ce 5:28.3.2-1~ubuntu.25.04~plucky
- **Python:** python3 3.13.3-1
- **PyTorch:** 2.7.1+cpu
- **SentenceTransformers:** 5.0.0
- **VectorDB:** qdrant/qdrant:v1.14.1
- **VectorDB model:** sentence-transformers/msmarco-distilbert-base-v4

Configuring Ubuntu 25.04

After installation, perform these configuration steps. We assume the login for the non-root user is `ptuser`.

1. Enable password-less sudo:

```
echo "$USER ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/$USER
sudo chmod 640 /etc/sudoers.d/$USER
```

2. Set the appropriate time zone:

```
sudo timedatectl set-timezone America/New_York
```

3. Extend the root LVM filesystem, if necessary:

```
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv
sudo resize2fs /dev/ubuntu-vg/ubuntu-lv
```

4. Disable unattended package updates:

```
sudo systemctl stop unattended-upgrades.service
sudo systemctl disable unattended-upgrades.service
sudo sed -i 's/:Unattended-Upgrade "1";/:Unattended-Upgrade "0";/' /etc/apt/apt.conf.d/20auto-upgrades
```

5. Install the latest updates:

```
sudo apt update
sudo apt upgrade
```

6. Install the standard Ubuntu packages you will need in subsequent steps:

```
sudo apt install ca-certificates curl wget lsb-release sysstat smartmontools vim nmon
numactl ipmitool tmux
```

7. Reboot the system:

```
sudo shutdown -r now
```

Installing Docker on Ubuntu 25.04

1. Remove any previous Docker packages:

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 \
    podman-docker containerd runc; do \
    sudo apt remove $pkg
done
```

2. Add Docker's official GPG key:

```
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

3. Add the Docker repository to the system:

```
echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.
    docker.com/linux/ubuntu \
    $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
    sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt update
```

4. Install Docker CE:

```
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

5. Add current user to Docker permissions group, and add Docker group to the user's current session:

```
sudo usermod -aG docker $USER
newgrp docker
```

6. Confirm Docker function under non-root user:

```
docker run --rm hello-world
```

7. Enable Docker services:

```
sudo systemctl enable --now docker.service
sudo systemctl enable --now containerd.service
```

Installing vector database (qdrant)

1. Create storage for the database and its index:

```
sudo mkdir -p /data/qdrant/qdrant_storage
sudo chown -R $USER:$USER /data/qdrant
```

2. Start the Docker script using the start script in the Appendix. This also automatically downloads the appropriate version of the Qdrant Docker container the first time it is run:

```
./start_qdrant.sh
```

Preparing the ingest environment

Ingesting Airbnb data into the vector database

1. Create a new directory on the SUT:

```
mkdir ingest
cd ingest
```

2. Create and activate a new virtual Python environment:

```
sudo apt install python3-venv
python3 -m venv .ingest
. .ingest/bin/activate
```

3. Install CPU version of PyTorch:

```
pip3 install -U torch --index-url https://download.pytorch.org/whl/cpu
```

4. Install Sentence Transformers:

```
pip3 install -U sentence-transformers
```

5. Install Qdrant client:

```
pip3 install -U qdrant-client
```

6. Install additional Python packages needed by ingestion script:

```
pip3 install -U psutil
```

7. Refer to the Appendix for a full list of all package versions used.

Preparing data for ingest

Note: The dataset is a proprietary source file from the PTChatterly benchmark. It's encoded in JSON format to be easily imported and parsed. We vectorized the description field, stripped some of the fields, and added the rest as text to the Vector DB. See the Appendix for a sample entry.

1. Copy the dataset to the ingest folder:

```
cp AirbnbProps-20240830.json.gz ingest/
```

2. Create a symbolic link to the property listings file dataset:

```
ln -s AirbnbProps-20240830.json.gz AirbnbProps.json.gz
```

Note: The benchmark strips several fields from the raw data before embedding and importing into the VectorDB. Here are the fields included in the final data (embedding was performed on the "description" field):

```
"name", "description", "notes", "property_type", "room_type", "country",
"state", "city", "market", "zipcode", "neighbourhood", "street",
"accommodates", "bedrooms", "beds", "bed_type", "bathrooms", "square_feet",
"price", "weekly_price", "monthly_price", "security_deposit", "cleaning_fee",
"maximum_nights", "minimum_nights", "first_review", "last_review",
"number_of_reviews", "access", "cancellation_policy", "house_rules",
"interaction", "amenities", "experiences_offered", "neighborhood_overview", "transit"
```

Running the benchmark

Ingesting Airbnb data into the vector database

1. Create a new directory on the SUT.
2. Run the ingest script:

```
python3 ingest_airbnb_qdrant_cpu.py -l -d cpu -a cpu -y bfloat16 -r 3 -b 1
python3 ingest_airbnb_qdrant_cpu.py -l -d cpu -a cpu -y float32 -r 3 -b 1
```

3. For a detailed list of options, use the following command (see the Outputs section for the output):

```
python3 ingest_airbnb_qdrant_cpu.py -h
```

Appendix

ingest_airbnb_qdrant_cpu.py -h output

```
Displaying Help
-d, --device Specify embedding model's compute device:
    "cuda" for GPUs, "cpu" for CPUs; Default: "cuda".

-y, --dtype Specify model precision:
    Possible values: float32, float16, bfloat16, or auto. Default: float32

-a, --affinity Specify CPU affinity strategy for multi-process embedding:
    Possible values: cpu, numa, cache, or None; Default: None
    "numa" : Pin each process to a numa node
    "cache": Pin each process to a cpu cache (L3 cache by default)
    "cpu"  : Pin each process to an individual logical cpu, starting at cpu 0

-p, --processes Specify number of multi-process targets:
    Integer value specifying number of processes; default: None
    For -a cpu, defaults to number of physical cores
    For -a cache, defaults to number of caches
    For -a numa, defaults to number of NUMA nodes
    If target_device is None and CUDA/NPU is available, then all available CUDA/NPU
devices will be used.
    If target_device is None and CUDA/NPU is not available, and no affinity is selected, then 4 CPU
processes will be used.
    Use -t to set number of threads per process (default is number of cores)

-m, --multi-process Specify whether to use multi-process embedding:
    "true" for multi-process, "false" for single-process; Default: "True"

-t, --omp-num-threads Specify number of threads per process (OMP_NUM_THREADS):
    Integer value specifying number of threads; default on this system: 128
    (if processes is unspecified, this will be split up)

-b, --batch-size Specify batch size(s) for embedding:
    Comma-separated list of integer values specifying batch sizes; default: [32]

-r, --runs Specify number of runs to execute:
    Integer value specifying number of runs; default: 1

-e, --expand Specify factor to expand corpus data size:
    Positive integer value; default: 1

-l, --local Use local model files only (no downloading):
    No value needed; if specified, sets local_files_only=True; Default: False

-c, --records Specify maximum number of records to process:
    Positive integer value; default: None (process all records)

-s, --stride Use CPU stride for CPU affinity (only applicable with -a cpu):
    No value needed; if specified, sets use_cpu_stride=True; Default: False
```

Script: start_qdrant.sh

```
#!/bin/bash

volume=/data/qdrant/qdrant_storage
log_level=info # [critical|error|warning|info|debug|trace], default: info
gpus=
#gpus='--gpus="device=0"'

name="${1:-vectordb-server}"
tag=${2:-v1.14.1}
port=${3:-6334}
device=${4:-cpu} # options: anything, amd, nvidia
threads=${5:-0}
numapin=${6:-0}
numanode=${7:-0}

image=qdrant/qdrant:"${tag}"

if [ "${device}" == "nvidia" ]; then
    image=qdrant/qdrant:"${tag}-gpu-nvidia"
elif [ "${device}" == "amd" ]; then
    image=qdrant/qdrant:"${tag}-gpu-amd"
fi

# Cleanup
echo "Stopping containers using the same ports."
docker stop `docker ps -f "expose=${port}" | awk '{if(NR>1)print $1}'` > /dev/null 2>&1
echo "Cleaning up old containers with the same name."
docker rm -f `docker ps -a -f "name=${name}" | awk '{if(NR>1)print $1}'` > /dev/null 2>&1
echo

# No NUMA pinning
if [ $numapin -eq 0 ]; then
    echo "Starting image ${image}, with name ${name}, on port ${port}"
    docker run -d --name "${name}" -v "${volume}:/qdrant/storage" -e QDRANT__TELEMETRY_DISABLED=true \
        -p 6333:6333 -p ${port}:6334 --cpus=128 ${gpus} "${image}"
else
    node=${numanode}
    cpulist=(cat /sys/devices/system/node/node${node}/cpulist)
    echo "Starting image ${image}, with name ${name}, on port ${port}, on numa_node ${node}, and cpulist
    ${cpulist}"
    numactl --physcpubind=${cpulist} --membind=${node} docker run -d --name "${name}" -v
    "${volume}:/qdrant/storage" -v "${config}:/qdrant/config/config.yaml" -e QDRANT__TELEMETRY_DISABLED=true \
        -p 6333:6333 -p ${port}:6334 ${gpus} --cpuset-mems=${node} --cpuset-cpus=${cpulist} \
        "${image}"
fi

docker ps --no-trunc -f "name=${name}"
```


Pip package list

Package	Version
annotated-types	0.7.0
anyio	4.9.0
certifi	2025.7.14
charset-normalizer	3.4.2
filelock	3.13.1
fsspec	2024.6.1
grpcio	1.73.1
h11	0.16.0
h2	4.2.0
hf-xet	1.1.5
hpack	4.1.0
httpcore	1.0.9
httpx	0.28.1
huggingface-hub	0.33.4
hyperframe	6.1.0
idna	3.10
Jinja2	3.1.4
joblib	1.5.1
MarkupSafe	3.0.2
mpmath	1.3.0
networkx	3.3
numpy	2.3.1
packaging	25.0
pillow	11.3.0
pip	25.0
portalocker	3.2.0
protobuf	6.31.1
psutil	7.0.0
pydantic	2.11.7
pydantic_core	2.33.2
PyYAML	6.0.2
qdrant-client	1.15.0
regex	2024.11.6
requests	2.32.4
safetensors	0.5.3
scikit-learn	1.7.1
scipy	1.16.0
sentence-transformers	5.0.0
setuptools	70.2.0
sniffio	1.3.1
sympy	1.13.3
threadpoolctl	3.6.0
tokenizers	0.21.2
torch	2.7.1+cpu
tqdm	4.67.1
transformers	4.53.2
typing_extensions	4.12.2
typing-inspection	0.4.1

Sample dataset

```
[
  {
    "name": "Luxurious 3 bedroom, centrum, 180m2",
    "summary": "This luxurious apartment is situated in the centrum, just a 5-minute walk from Leidseplein. The modern apartment has 2 floors, 3 verdroemd and free WIFI. A wide variety restaurants and cafes is located in the direct surroundings, Rijksmuseum 10min walk and tramstop is 200 metres away.",
    "space": "In the heart of hip & lively Amsterdam Centrum Oud-West lies our sunny and spacious apartment. This luxurious apartment is situated perfectly in the Helmerbuurt neighbourhood, just a 5-minute walk from Leidseplein, Vondelpark and the Museum area. You will stay in the middle of Amsterdam but will have a good night sleep as the neighborhood is very peaceful. The location of the apartment in combination to the space that it offers is quite unique! Amsterdam city centre is your backyard! The modern 180 m2 apartment has 2 floors and a rooftop terrace. On the lower floor is the light and spacious livingroom, a fully equipped kitchen and a separate toilet in the hallway. Ofcourse you will have wifi access, flat screens tv's and apple tv. The upper floor features 3 beautiful bedrooms; 2 of them with an
```

```

amazing kingsize bed and the third with one single. The bathroom has a separate shower and a great bath
and shampoo and soap is always provided! In the hallway is the 2nd separate toilet. You",
  "description": "This luxurious apartment is situated in the centrum, just a 5-minute walk from
Leidseplein. The modern apartment has 2 floors, 3 verdroomd and free WIFI. A wide variety restaurants and
cafes is located in the direct surroundings, Rijksmuseum 10min walk and tramstop is 200 metres away.",
  "experiences_offered": "none",
  "neighborhood_overview": "There is a wide variety of cafes and restaurants is located in the direct
surrounding of the apartment and the tram stop (which is 100 metres away) offers direct links to Anne
Frank House, Dam Square and Central Station. Getting around is as easy as it can be from here. You can
literally walk anywhere which is the most fun from my point of view, but you can also rent a bike from
the bike rental downstairs. Getting a taxi is also easy and all lines of public transport are just
around the corner.",
  "notes": "Perfect location to enjoy Amsterdam's famous events like Gay Parade, Sail, IBC and
Amsterdam Dance Events!",
  "transit": null,
  "access": "",
  "interaction": "",
  "house_rules": "I expect you to treat my beloved apartment as you would like others to treat your
own house and we will be fine! Please respect my lovely neighbors and neighbourhood...",
  "host_response_time": "within a day",
  "host_response_rate": 50.0,
  "host_acceptance_rate": null,
  "host_neighbourhood": "Oud-West",
  "host_listings_count": 1.0,
  "host_total_listings_count": 1.0,
  "host_verifications": [
    "email",
    "phone",
    "facebook",
    "reviews",
    "jumio"
  ],
  "street": "Oud-West, Amsterdam, North Holland 1054 AA, Netherlands",
  "neighbourhood": "Oud-West",
  "neighbourhood_cleansed": "De Baarsjes - Oud-West",
  "neighbourhood_group_cleansed": "",
  "city": "Amsterdam",
  "state": "North Holland",
  "zipcode": "1054 AA",
  "market": "Amsterdam",
  "smart_location": "Amsterdam, Netherlands",
  "country_code": "NL",
  "country": "Netherlands",
  "latitude": 52.3652366041,
  "longitude": 4.8782495751,
  "property_type": "Apartment",
  "room_type": "Entire home apt",
  "accommodates": 6.0,
  "bathrooms": 1.0,
  "bedrooms": 3.0,
  "beds": 3.0,
  "bed_type": "Real Bed",
  "amenities": [
    "TV",
    "Internet",
    "Wireless Internet",
    "Kitchen",
    "Pets allowed",
    "Indoor fireplace",
    "Heating",
    "Family/kid friendly",
    "Smoke detector",
    "First aid kit",
    "Fire extinguisher",
    "Essentials",
    "Shampoo",
    "Hair dryer",
    "Iron",
    "Laptop friendly workspace"
  ],
  "square_feet": null,
  "price": 600.0,

```

```

    "weekly_price": null,
    "monthly_price": 6000.0,
    "security_deposit": 500.0,
    "cleaning_fee": 50.0,
    "guests_included": 4.0,
    "extra_people": 50.0,
    "minimum_nights": 2.0,
    "maximum_nights": 1125.0,
    "has_availability": null,
    "number_of_reviews": 31.0,
    "first_review": "2015-08-02",
    "last_review": "2016-11-27",
    "review_scores_rating": 89.0,
    "review_scores_accuracy": 9.0,
    "review_scores_cleanliness": 8.0,
    "review_scores_checkin": 9.0,
    "review_scores_communication": 10.0,
    "review_scores_location": 10.0,
    "review_scores_value": 9.0,
    "license": null,
    "jurisdiction_names": "Amsterdam",
    "cancellation_policy": "strict",
    "calculated_host_listings_count": 1.0,
    "reviews_per_month": 1.52,
    "features": [
      "Host Has Profile Pic",
      "Host Identity Verified",
      "Is Location Exact"
    ],
    "lang": "en",
    "geolocation": null
  },

```

[Read the report ▶](#)

This project was commissioned by Dell Technologies.



Facts matter.®

Principled Technologies is a registered trademark of Principled Technologies, Inc.
All other product names are the trademarks of their respective owners.

DISCLAIMER OF WARRANTIES; LIMITATION OF LIABILITY:

Principled Technologies, Inc. has made reasonable efforts to ensure the accuracy and validity of its testing, however, Principled Technologies, Inc. specifically disclaims any warranty, expressed or implied, relating to the test results and analysis, their accuracy, completeness or quality, including any implied warranty of fitness for any particular purpose. All persons or entities relying on the results of any testing do so at their own risk, and agree that Principled Technologies, Inc., its employees and its subcontractors shall have no liability whatsoever from any claim of loss or damage on account of any alleged error or defect in any testing procedure or result.

In no event shall Principled Technologies, Inc. be liable for indirect, special, incidental, or consequential damages in connection with its testing, even if advised of the possibility of such damages. In no event shall Principled Technologies, Inc.'s liability, including for direct damages, exceed the amounts paid in connection with Principled Technologies, Inc.'s testing. Customer's sole and exclusive remedies are as set forth herein.