Readers Are The Leader

INSTANCE TYPE	GPU TYPE	GPU MEMORY	SUITABLE FOR LLMS
g5.2xlarge	NVIDIA A10G	24 GB	Yes, supports Llama 3.1 (8B model)
g4dn.xlarge	NVIDIA T4	16 GB	Yes, suitable for smaller models
p3.2xlarge	NVIDIA V100	16 GB	Good for larger models and training
p3.8xlarge	NVIDIA V100	32 GB	Excellent for high-performance tasks
p4d.24xlarge	NVIDIA A100	40 GB	Best for large-scale models and training

| **Provider | **GPU Options** | **Driver Information** | **Download Link** |**

|--------------|------------------|----------------------|--------------------||--------------|---------------|----------------------|--------------------|

| **AWS** | NVIDIA A10G, A100, H100 | NVIDIA GPU Driver Extension for AWS | [NVIDIA Drivers](https://www.nvidia.com/Download/index.aspx) |

| **GCP** | NVIDIA A100, V100, T4 | NVIDIA GPU Driver for GCP | [NVIDIA Drivers](https://www.nvidia.com/Download/index.aspx) |

| **Azure** | NVIDIA A10, A100, H100, AMD MI25 | NVIDIA GPU Driver Extension for Azure | [NVIDIA Drivers](https://www.nvidia.com/Download/index.aspx) |

| **Apple** | Apple M1, M2 GPUs | Apple Graphics Driver Updates | [Apple Support](https://support.apple.com) |

| **NVIDIA** | A10, A100, H100, V100 | NVIDIA Drivers for various GPUs | [NVIDIA Drivers](https://developer.nvidia.com/cuda-downloads) |

Check The Leader: https://docs.nvidia.com/datacenter/tesla/driver-installation-guide/#switching-between-driver-module-flavors

| **AMD** | Radeon Instinct MI25, MI300X | AMD GPU Drivers | [AMD Drivers] (https://www.amd.com/en/support) |

| **Intel** | Integrated Intel Graphics, Intel NPU | Intel Graphics Drivers | [Intel Drivers](https://downloadcenter.intel.com) |

| **Oracle** | NVIDIA A100, H100, AMD MI300X | NVIDIA and AMD Drivers for OCI | [NVIDIA Drivers] (https://www.nvidia.com/Download/index.aspx), [AMD Drivers] (https://www.amd.com/en/support) |

Details on Each Provider

AWS
AWS offers a range of NVIDIA GPUs, including the **A10G**, **A100**, and **H100**. The NVIDIA GPU Driver Extension can be installed to manage these GPUs effectively.

GCP
Google Cloud Platform provides access to NVIDIA GPUs like the **A100**, **V100**, and **T4**. The drivers can be installed via the NVIDIA GPU Driver for GCP.

Azure
Microsoft Azure supports NVIDIA GPUs such as the **A10**, **A100**, and **H100**, as well as AMD's **MI25**. The NVIDIA GPU Driver Extension is available for installation.

Apple
Apple's M1 and M2 chips come with integrated GPUs. Driver updates can be found on the Apple Support page.

NVIDIA
NVIDIA provides a variety of GPUs for AI workloads, including the **A10**, **A100**, **H100**, and **V100**. Drivers can be downloaded from the NVIDIA website.

AMD
AMD offers GPUs like the **Radeon Instinct MI25** and **MI300X** for AI applications. Drivers are available on the AMD support page.

Intel
Intel provides integrated graphics and NPU options. Drivers can be downloaded from Intel's official site.

Oracle
Oracle Cloud Infrastructure supports NVIDIA and AMD GPUs for AI workloads. Drivers for these GPUs can be found on the NVIDIA and AMD websites.

This table summarizes the key options available for AI model deployments across major cloud providers and hardware manufacturers, along with links to download the necessary drivers.
——

Verify NVIDIA Drivers and CUDA:

wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2404/x86_64/cuda-keyring_1.1-1_all.deb

sudo dpkg -i cuda-keyring_1.1-1_all.deb

sudo apt-get update

sudo apt-get -y install cuda-toolkit-13-0

sudo apt-get install -y cuda-drivers

cat /usr/local/cuda/version.json

Check GPU status and driver:

nvidia-smi
nvcc --version

Real-time monitoring:

watch -n 1 nvidia-smi

Check NVIDIA kernel modules:

lsmod | grep nvidia

Check GPU memory:

nvidia-smi --query-gpu=memory.total,memory.used,memory.free --format=csv

Set Env Variable:

export OLLAMA_ENABLE_GPU=1

——

 # Increase file descriptors Linux Kernel Tuning

```
sudo nano /etc/sysctl.conf

# File descriptor limits
fs.file-max=2097152
fs.nr_open=2097152

# Virtual memory tuning
vm.swappiness=10
vm.max_map_count=262144
vm.dirty_ratio=10
vm.dirty_background_ratio=5
vm.overcommit_memory=1
vm.overcommit_ratio=90

# Process/thread limits
kernel.pid_max=4194303
kernel.threads-max=2097152
```

```
# Network tuning
net.core.somaxconn=65535
net.core.netdev_max_backlog=250000
net.core.rmem_max=268435456
net.core.wmem_max=268435456
net.core.optmem_max=67108864
net.ipv4.tcp_fin_timeout=15
net.ipv4.tcp_keepalive_time=300
net.ipv4.tcp_keepalive_intvl=30
net.ipv4.tcp_keepalive_probes=5
net.ipv4.tcp_max_syn_backlog=3240000
net.ipv4.tcp_max_tw_buckets=1440000
net.ipv4.tcp_rmem=4096 87380 268435456
net.ipv4.tcp_wmem=4096 65536 268435456
net.ipv4.tcp_tw_reuse=1
net.ipv4.ip_local_port_range=1024 65535
net.ipv4.tcp_mtu_probing=1

# Shared memory
kernel.shmmax=68719476736
kernel.shmall=4294967296
```

Persist settings

sudo sysctl -p
sudo sysctl --system
ulimit -n 1048576

Verify all settings in one command

```
grep -v '^#' /etc/sysctl.conf | awk -F= '{print $1}' | while read param; do
    echo -n "$param = "
    sysctl $param
done
```

Install Ollama: https://ollama.com/

sudo snap install ollama
ollama --version

Pull and run models

https://ollama.com/danielsheep/gpt-oss-20b-Unsloth

With Quantization

ollama pull danielsheep/gpt-oss-20b-Unsloth:latest
ollama run danielsheep/gpt-oss-20b-Unsloth:latest

ollama pull NeuroEquality/neuralquantum-coder
ollama run NeuroEquality/neuralquantum-coder

ollama pull haghiri/DeepSeek-V3-0324:IQ1_S
ollama run haghiri/DeepSeek-V3-0324:IQ1_S

With out Quantization

ollama pull gpt-oss
ollama run gpt-oss

ollama pull deepseek-r1:671b
ollama run deepseek-r1:671b

ollama pull gemma3:4b
ollama run gemma3:4b

Other useful commands:

```
ollama list        # List installed models
ollama show gpt-oss # Show model details
ollama rm gpt-oss    # Remove model
ollama serve       # Start Ollama API server
```

————————————————————————

Using REST API: Integrate Ollama into Applications

```
curl http://localhost:11434/api/generate -d '{
  "model": "gpt-oss",
  "prompt": "Summarize quantum entanglement in 3 sentences."
}'
```

==========

Install Docker: Docker Web UI Integration

sudo snap install docker

Run Open-WebUI container:

```
sudo docker run -d \
  --network host \
  --name open-webui \
  -p 3000:8080 \
  -e OLLAMA_BASE_URL=http://localhost:11434 \
  -v open-webui:/app/backend/data \
  --add-host=host.docker.internal:host-gateway \
  --restart always \
  ghcr.io/open-webui/open-webui:main
```

Access the Inference:

```
curl -s https://api.ipify.org
dig +short myip.opendns.com @resolver1.opendns.com
```

```
curl -s https://ipinfo.io/ip
curl ifconfig.me
```

h*ttp://13.234.120.211:8080/*

————————————————————————

Real-time monitoring: Monitor GPU Usage

watch -n 1 nvidia-smi

Check NVIDIA kernel modules:

lsmod | grep nvidia

-============================————+++++++++++++++++++++
++++++++++

Troubleshooting

# Docker container management
sudo docker stop open-webui
sudo docker start open-webui
sudo docker rm -f open-webui

Check Ollama port:

sudo ss -tnlp | grep 11434

Check API tags:

curl http://localhost:11434/api/tags

Free up storage:

ollama rm deepseek-r1:8b

Stop/start Ollama service

sudo snap stop ollama
sudo snap start ollama
