**AIfinity Research**

*Your GenAI Launchpad ! ( https://aifinity.online)*

**Amazon EC2 Instance Setup for Hosting WAN 2.1 Models**

**Purpose** : Step-by-Step guide to configure AWS EC2 g5.2xlarge CPU with NVIDIA AMI *Deep Learning OSS Nvidia Driver AMI GPUPyTorch 2.7 (Ubuntu 22.04)*

**About WAN Models** : WAN, particularly the Wan2.1 video generation model, was developed and open-sourced by Alibaba in early 20252. It’s part of their broader push into generative AI, and they’ve released it under an open-source license to encourage community use and innovation.

* **Wan-AI/Wan2.1-T2V-1.3B**

This is a **dedicated Text-to-Video model** from the Wan2.1 family. Key highlights:

* Generates 480p videos in ~4 minutes on an RTX 4090
* Requires only 8.19 GB VRAM
* Supports English and Chinese text prompts
* Built on a 3D Spatio-Temporal VAE, preserving temporal coherence
* **WAN VACE 1.3B & 14B**

These are part of the **Wan2.1 VACE** (Video Anything Creation Engine) suite, designed for **all-in-one video generation and editing**. They support tasks like:

* Text-to-Video (T2V)
* Image-to-Video (I2V)
* Video-to-Video (V2V)
* Reference-to-Video (R2V)
* **Masked Video Editing (MV2V)**

**1.3B** is optimized for **consumer-grade GPUs** (≈8 GB VRAM), while **14B** targets **high-end setups** (24 GB+ VRAM), offering higher fidelity and resolution (up to 720p or more)

WAN Models are available at Huggingface

<https://huggingface.co/Wan-AI>

Github Repo

<https://github.com/ali-vilab/VACE>

**Real World Applications**

* AI-generated short films: Artists and studios can generate entire video sequences from text prompts or storyboards.
* Video editing & enhancement: Automatically upscale, color-correct, or modify scenes using text or image guidance.
* Multilingual video content: Generate videos with embedded captions in multiple languages—great for global audiences.

🛍️ E-commerce & Marketing

* Product showcase videos: Turn product images into dynamic 360° or lifestyle videos.
* Ad creatives: Generate personalized video ads based on user behavior or regional preferences.

🏫 Education & Training

* Interactive learning modules: Create visual explanations of complex topics (e.g., physics simulations or historical reenactments).
* Language learning: Generate scenario-based videos for immersive language practice.

🏗️ Engineering & Design (this one’s right up your alley!)

* CAD-to-video visualization: Animate product designs or tool mechanisms for client presentations or internal reviews.
* Simulation of assembly processes: Generate instructional videos from CAD models to train technicians or customers.

🧠 AI Coaching & Content Creation

* Tutorial generation: Produce engaging video tutorials for your AI coaching program.
* Dynamic course content: Auto-generate visual examples or explainer videos based on lesson plans or user queries.

📰 Journalism & Social Media

* News summarization: Turn headlines or articles into short video summaries.
* Social storytelling: Creators can generate stylized video content from a single image or phrase.

**Deploying WAN Models to AWS EC2 – Why ?**

Deploying WAN models like Wan2.1 on private AWS EC2 instances offers companies a powerful blend of performance, control, and scalability—especially when working with compute-heavy tasks like video generation. Here's how it helps:

🔐 1. Data Privacy & Security

Running models on private EC2 instances ensures sensitive data never leaves your environment. This is crucial for industries like healthcare, finance, or defense where compliance and confidentiality are non-negotiable.

⚙️ 2. Customization & Fine-Tuning

You can fine-tune WAN models on proprietary datasets or integrate them into custom pipelines—something that’s often limited or restricted on public platforms.

🚀 3. Scalable Performance

Using GPU-optimized EC2 instances (like g5.xlarge with NVIDIA A10G) allows for high-performance video generation without investing in physical infrastructure. You can scale up or down based on demand.

💸 4. Cost Efficiency

With pay-as-you-go pricing, companies avoid the upfront costs of on-prem hardware. Spot instances or reserved pricing can further reduce expenses.

🧩 5. Seamless Integration

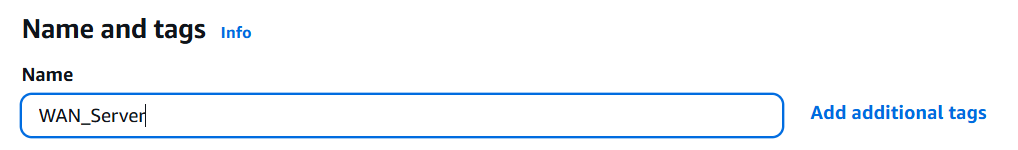
Deploying on AWS makes it easier to integrate with other cloud-native services—like S3 for storage, Lambda for automation, or SageMaker for orchestration.

🛠️ 6. DevOps & Automation

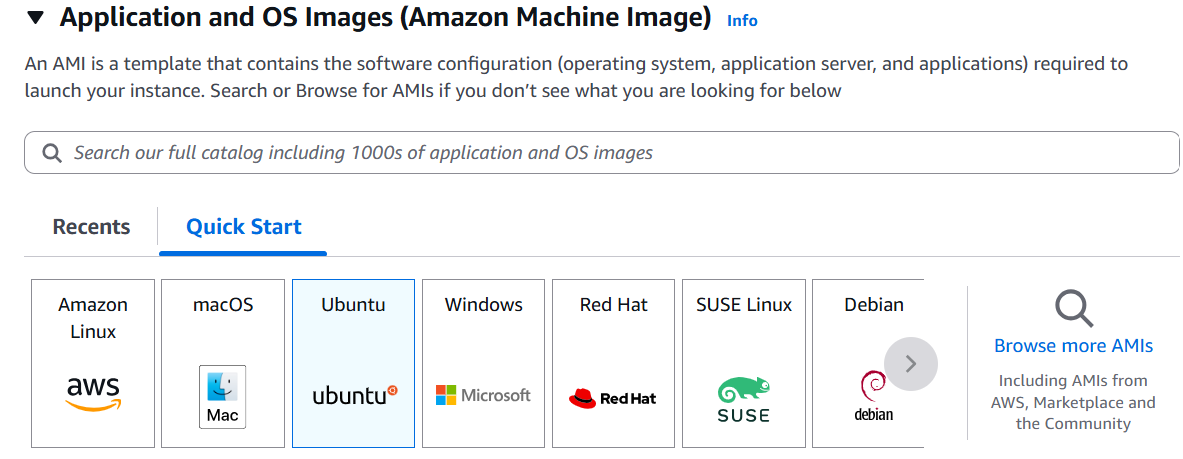
You can automate deployment, monitoring, and updates using tools like Terraform, CloudWatch, and CI/CD pipelines, making WAN models production-ready.

**AWS EC2 Creation & Installation Steps**

* 1. Login/Sign up to AWS Console <https://aws.amazon.com/console/>
  2. Go to EC2
  3. Launch Instance --> Give it a name “WAN\_Server”

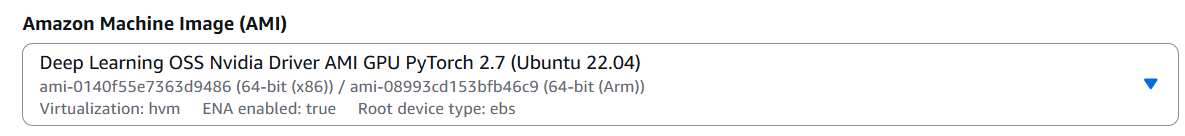


* 1. Application and OS Image . Choose Ubuntu

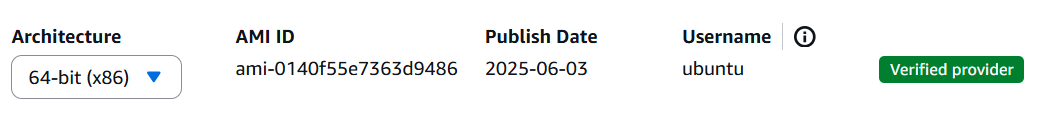
****

* 1. Choose Amazon Machine Image ( AMI ) – Deep Learning OSS Nvidia Driver AMI GPU

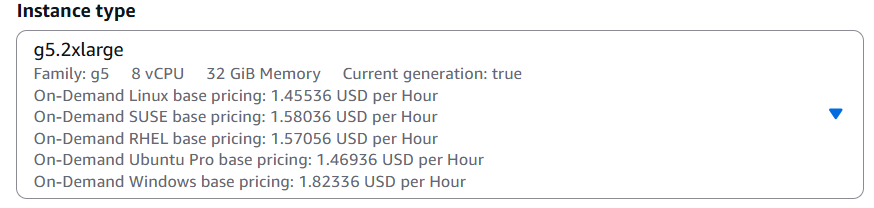
PyTorch 2.7 (Ubuntu 22.04)

****

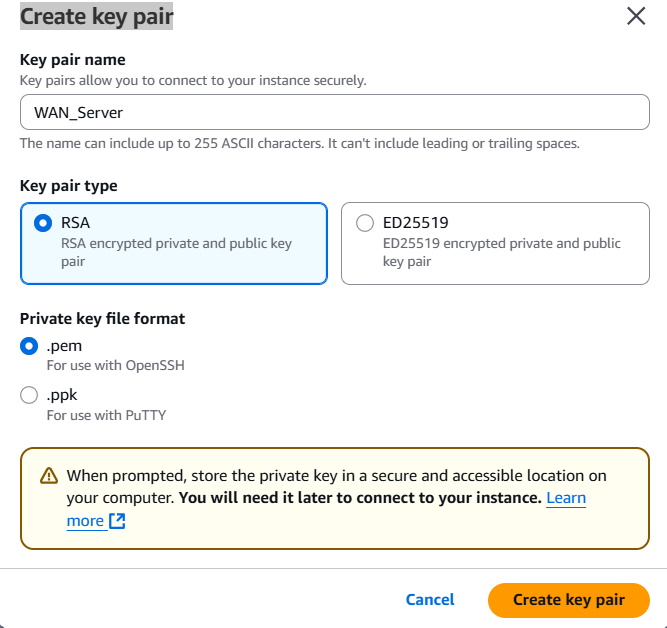
* 1. Architecture – Keep 64-bit ( x86)



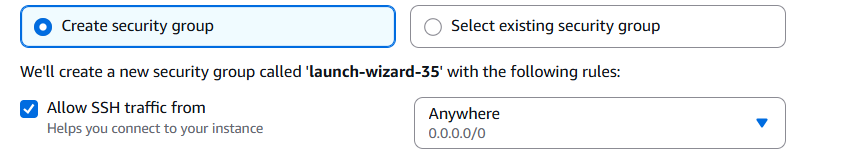
* 1. Instance Type – g5.2xlarge

****

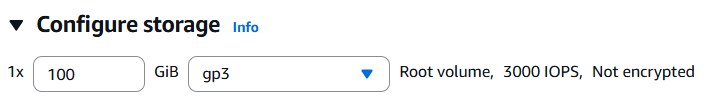
8 Create new key pair – “WAN\_Server” , type .pem ( Don’t forget to download , save it in a safe location )



9 Network Settings . Choose “Create Security Group” , “Allow SSH traffic from Anywhere 0.0.0.0“



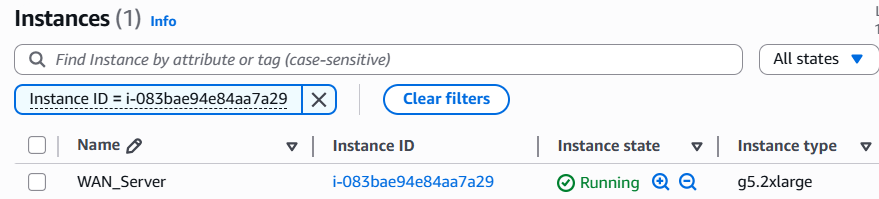
10 Configure Storage – make it 100 GB ( SSD)



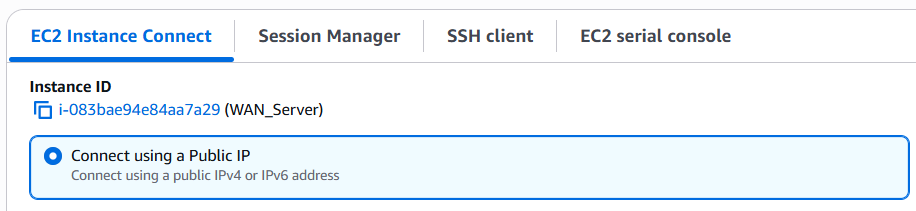
*Note : to deploy larger models like WAN 2.1 14B , you have to choose higher SSD size as per the model size*

11 Click on Launch Instance. Wait for a minute for the instance to spin up

12 Go to Instances tab -> Click on the instance



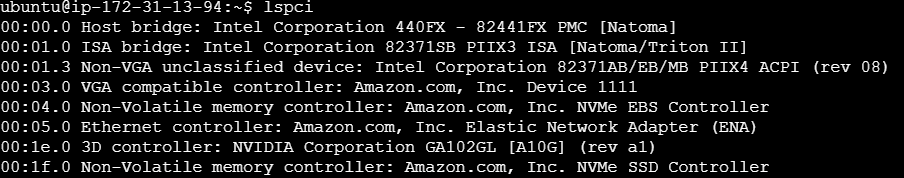
13 Choose “EC2 Instance Connect” and “Connect using a public IP”



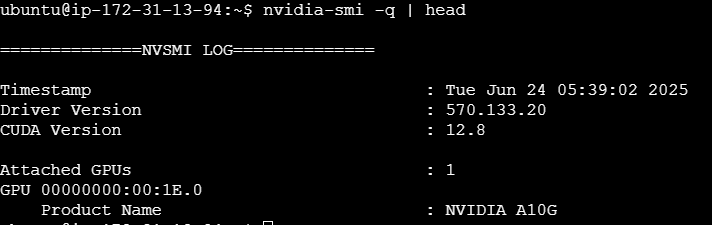
14 Wait for prompt to open up

15 Execute commands

CMD:/>lspci # to check if GPU is running



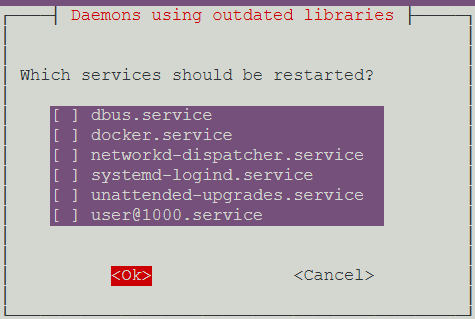
CMD:/>nvidia-smi -q | head # to check if CUDA drivers are installed



16 Install Packages

*sudo apt update && sudo apt upgrade -y && sudo apt install -y git*

*When prompted with below dialog box, use tab to select “Ok” and Press Enter*

**

17 Download VACE code base

git clone <https://github.com/ali-vilab/VACE.git>

cd VACE

18 Re-install Torch

pip install torch==2.5.1 torchvision==0.20.1 --index-url <https://download.pytorch.org/whl/cu124>

pip install -r requirements.txt

19 Re-install flash-attn ( for NVIDIA updated packages )

*pip install flash-attn --no-build-isolation --extra-index-url* [*https://pypi.nvidia.com*](https://pypi.nvidia.com)

20 Add bin to PATH & reload .bashrc

*echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc*

*source ~/.bashrc*

21 Install WAN2.1 code base

pip install [wan@git+https://github.com/Wan-Video/Wan2.1](mailto:wan@git+https://github.com/Wan-Video/Wan2.1)

22 Install Huggingface CLI

pip install huggingface\_hub[cli]

huggingface-cli login ( paste the HF\_TOKEN)

Note : When prompted with “Add token to git credentials” – Click No

23 Download you model – here Wan2.1-VACE-1.3B

huggingface-cli download Wan-AI/Wan2.1-VACE-1.3B --local-dir ./models/Wan2.1-VACE-1.3B

*24 Check the disk size of model ( a Whooping 18 GB …)*

du -sh ./models/Wan2.1-VACE-1.3B

25 *Test your model in EC2 first – with 1 image as input & a prompt*

TRYING THIS - 1 IMAGES

python3 vace/vace\_wan\_inference.py \

--model\_name vace-1.3B \

--ckpt\_dir ./models/Wan2.1-VACE-1.3B \

--prompt "A little girl in green spring clothes plays joyfully in a festive setting." \

--src\_ref\_images assets/images/girl.png

--size 480p \

--frame\_num 81

*25A Test your model in EC2 first – with 2 images as input & a prompt*

python3 vace/vace\_wan\_inference.py \

--model\_name vace-1.3B \

--ckpt\_dir ./models/Wan2.1-VACE-1.3B \

--prompt "A little girl in red spring clothes plays joyfully with a cartoon snake in a festive setting." \

--src\_ref\_images assets/images/girl.png,assets/images/snake.png \

--size 480p \

--frame\_num 81

26 To test the generated image navigate to path

cd /home/ubuntu/VACE/results/

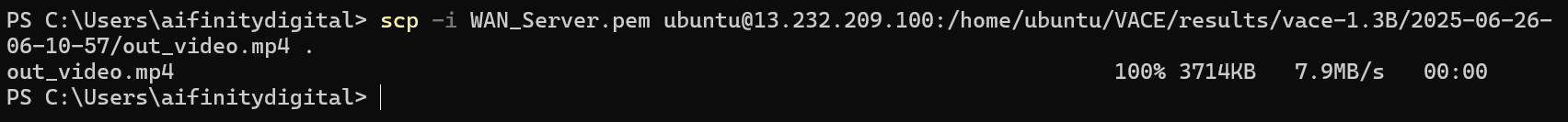
Verify if out\_video.mp4 is present in timestamped folder ( ex 2025-06-25-06-57-54)

27 To view the video in your PC , you need to use the .pem file created in **step** **8**

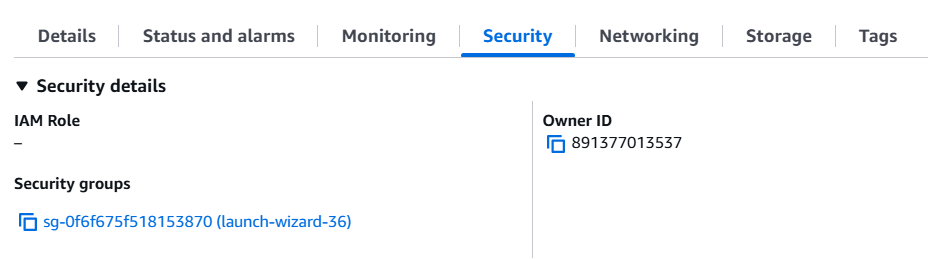
28 Copy the .pem file ( ex WAN\_Server.pem ) to user path . For ex C:/Users/aifinitydigital

29 Launch Windows Powershell and execute below command ( Copy your EC2 instance’s IP address from

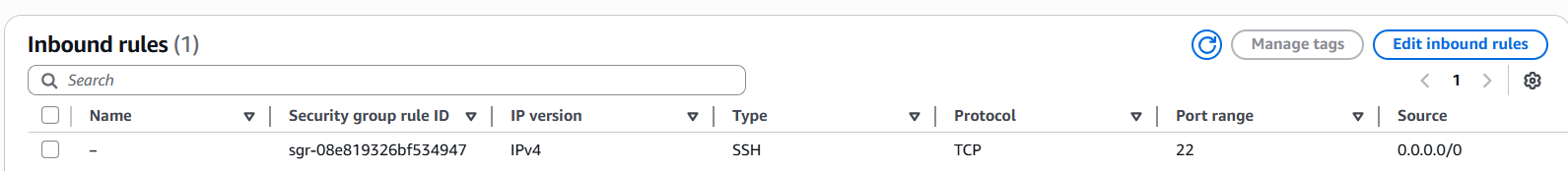
*scp -i WAN\_Server.pem ubuntu@13.232.209.100:/home/ubuntu/VACE/results/vace-1.3B/2025-06-26-06-10-57/out\_video.mp4 .*



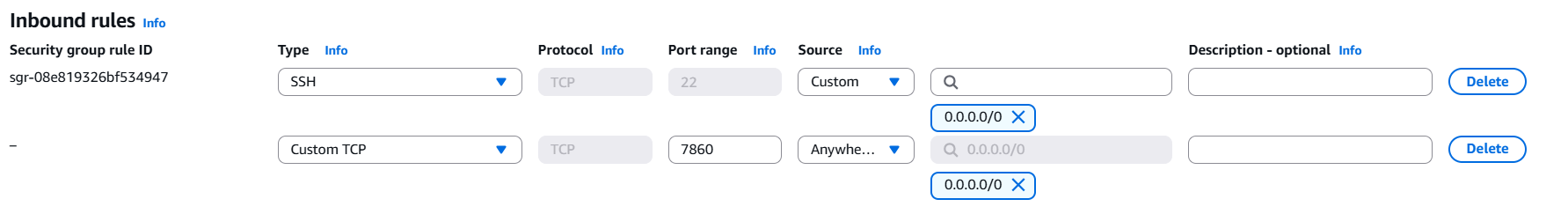
30 To enable Client Code to connect to EC2 , we have to expose TCP port in EC2 instamce . Go to EC2 in AWS Console 🡪 Select your ec2 instance 🡪 Security Tab 🡪Click on Security Group “sg-0f6..”



31 Click on “Edit Inbound Rules”

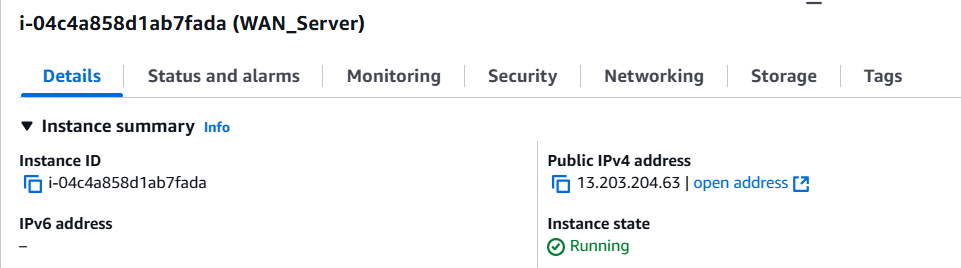


32 Add New Rule “Custom TCP” , port 7860 ( or 8000 ) , Source -> From Anywhere ( 0.0.0.0/0)



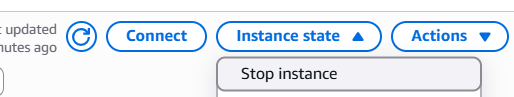
*Note : In production systems , its not recommended to expose your EC2 instance to public . The EC2 instances in production are always deployed in a private subnet with private IP only* . For demo purposes only EC2 is being exposed via public IP

33 Copy the Public IP address ( ex 13.203.204.63) of the ec2 and use it in the Client Code



Warning : Beware of keeping the EC2 in running state - which will attract cost . Remember g5.2xlarge CPU’s cost ~1.5 $ an hour . Always remember to keep it STOPPED when not in use

Go to EC2 Instance -> Instance State -> Stop Instance



Optional : Stopping an instance will also kill the server.py from running . So when you restart the instance , you have to use the command in step 29 to restart the uvicorn server OR you may configure a systemd service file ( refer APPENDIA A below )

**APPENDIX A**

**Step-by-step: Create a systemd service**

1. Open the service file

sudo nano /etc/systemd/system/wan\_server.service

1. Paste this configuration

[Unit]

Description=Start WAN2.1 FastAPI server with uvicorn

After=network.target

[Service]

WorkingDirectory=/home/aifinity/Wan2.1/demo\_text2video/server

ExecStart=/usr/bin/env nohup uvicorn server:app --host 0.0.0.0 --port 7860

StandardOutput=append:/home/aifinity/Wan2.1/demo\_text2video/server/server.log

StandardError=append:/home/aifinity/Wan2.1/demo\_text2video/server/server.log

Restart=always

User=aifinity

[Install]

WantedBy=multi-user.target

1. Reload and Restart

sudo systemctl daemon-reexec

sudo systemctl enable wan\_server.service

sudo systemctl start wan\_server.service

1. (Optional) Check that it's running:

sudo systemctl status wan\_server.service